

5-2019

Development of Robust Control Laws for Disturbance Rejection in Rotorcraft UAVs

Johannes Verberne

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerospace Engineering Commons](#)

Scholarly Commons Citation

Verberne, Johannes, "Development of Robust Control Laws for Disturbance Rejection in Rotorcraft UAVs" (2019). *Dissertations and Theses*. 447.
<https://commons.erau.edu/edt/447>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

DEVELOPMENT OF ROBUST CONTROL LAWS FOR DISTURBANCE
REJECTION IN ROTORCRAFT UAVS

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Johannes Verberne

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

May 2019

Embry-Riddle Aeronautical University

Daytona Beach, Florida

DEVELOPMENT OF ROBUST CONTROL LAWS FOR DISTURBANCE
REJECTION IN ROTORCRAFT UAVS

by

Johannes Verberne

A Thesis prepared under the direction of the candidate's committee Chairman, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE



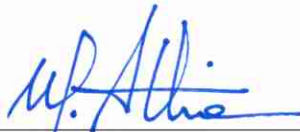
Chair, Dr. Hever Moncayo



Member, Dr. Richard Prazenica



Member, Dr. Eric Coyle



Graduate Program Coordinator, Dr. Magdy Attia



Dean of College of Engineering, Dr. Maj Mirmirani



Senior Vice President for Academic Affairs and Provost, Dr. Lon Moeller

4/22/2019

Date

4/22/19

Date

4/23/19

Date

ACKNOWLEDGMENTS

I would like to acknowledge those who supported me during the completion of this thesis.

Most importantly, I would like to thank my parents, Patricia and Ton, and my brothers, Fons, Ton, Jeff and Jim for their unlimited support, invaluable encouragements and unconditional love. All of you push me to be a better person every day. I'm very grateful to have your support. Ik hou van jullie allemaal.

My deepest thank you to my beautiful girlfriend Ana, for always being by my side even when times are tough. My accomplishments in academia and life would not be possible without your support, love and infinite patience. I love you dearly.

To my advisor Dr. Hever Moncayo for giving me the opportunity to work under his leadership. I will cherish the valuable lessons you taught me and the great discussions we had. Thank you for your enthusiasm and engagement throughout this research.

I would like to thank the committee members, Dr. Richard Prazenica and Dr. Eric Coyle for their support, insights and feedback towards the completion of this thesis.

To my friends at the Advanced Dynamics and Control Laboratory, Karina, Yomary, Nolan, Diana, Mansi, Sarath, Andrei and Ethan, thank you for your friendship. I will treasure the memories forever.

Finally, a thank you to the ERAU athletics department, especially coach David Paschal, for teaching me crucial life lessons through the sport of tennis.

This research effort has been supported by the Defense Advanced Research Projects Agency (DARPA).

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
SYMBOLS	xii
ABBREVIATIONS	xiv
ABSTRACT	xv
1. Introduction	1
1.1. Background.....	1
1.2. Research Objective	5
1.3. Thesis Outline.....	6
2. Literature Review	8
2.1. Stabilizing Control Architectures for Quadrotor UAVs.....	9
2.1.1. Linear control	9
2.1.2. Nonlinear control.	14
2.1.3. Adaptive control.....	19
2.1.4. Intelligent control.	23
2.2. Stability of Nonlinear Systems.....	28
2.2.1. General stability definitions	29
2.2.2. Showing stability of equilibrium points	30
2.2.3. Stability of adaptive and intelligent control systems	31
2.3. Discrete Time Systems	32
2.4. UAV External Disturbance Rejection Methodologies	34
2.4.1. Wind estimation methods.....	35
2.4.2. Wind rejection methods.....	36
3. System Mathematical Modeling	38
3.1. Quadrotor UAV Model.....	38
3.1.1. Kinematics.....	38
3.1.2. Rigid body dynamics	39
3.1.3. Control allocation.....	45
3.2. Wind Model.....	45
3.2.1. Modeling of the wind environment	45
3.2.2. Wind induced forces	48
3.2.3. Wind induced moments.....	52
4. Control Law Architectures	53
4.1. Baseline NLDI.....	53
4.1.1. Inner stability loop inversion.....	55
4.1.2. Outer tracking loop inversion.....	59
4.2. Extended NLDI with Wind Effects	64
4.2.1. Extended inner stability loop inversion	66

4.2.2. Extended outer tracking loop inversion	67
4.3. Adaptive ANN Augmentation	71
4.3.1. Adaptive ANN selection analysis	72
4.3.2. Adaptive ANN augmentation architecture	79
4.3.3. Stability analysis	82
4.3.4. Discrete adaptive ANN architecture	85
4.4. \mathcal{L}_1 Output-Feedback Adaptive Control Augmentation	86
4.4.1. \mathcal{L}_1 output-feedback adaptive control architecture	87
4.4.2. Stability analysis	92
4.4.3. \mathcal{L}_1 output-feedback adaptive control augmentation architecture	95
4.4.4. Discrete \mathcal{L}_1 output-feedback adaptive control architecture	100
5. Simulation Analysis	103
5.1. Quadrotor UAV Simulation Environment	104
5.1.1. DJI flamewheel F330 model	104
5.1.2. Sensor and filter model	106
5.1.3. Waypoint navigation and control architecture model	108
5.1.4. Wind model	110
5.2. Performance Metric	114
5.3. Performance Evaluation Baseline NLDI under Nominal Conditions	117
5.4. Performance Evaluation Case Study	120
5.4.1. Extended NLDI	123
5.4.2. Adaptive ANN augmentation	124
5.4.3. \mathcal{L}_1 output-feedback adaptive control augmentation	128
5.5. Monte Carlo Analysis	129
6. Quadrotor UAV Testbed & Implementation Environment	135
6.1. DJI Flamewheel F330 Research Testbed	136
6.1.1. Pixhawk flight computer and sensor package	138
6.1.2. Power and propulsion systems	141
6.2. Embedded Software	143
6.2.1. Pixhawk support package	144
6.2.2. S-function development	145
6.2.3. Flamewheel F330 flight code	146
6.3. Ground Weather Station	147
6.4. Flight Testing Locations	150
6.4.1. MicaPlex indoor unmanned systems testing facility	150
6.4.2. ERAU artificial turf softball field	152
7. Implementation Analysis	153
7.1. Performance Evaluation Indoor Nominal Conditions	155
7.1.1. Baseline NLDI	157
7.1.2. Adaptive ANN augmentation	157
7.1.3. \mathcal{L}_1 output-feedback adaptive control augmentation	162
7.2. Outdoor Performance Evaluation	163
7.2.1. Nominal wind disturbance conditions	163

7.2.2. Low wind disturbance conditions	166
7.2.3. Medium wind disturbance conditions.....	168
7.2.4. High wind disturbance conditions	174
8. Conclusions	177
9. Future Work & Recommendations	179
REFERENCES	182
Appendix.....	194

LIST OF TABLES

Table 2.1	The Application Development of ANN (Al-Mahasneh, Anavatti, & Garatt, 2017)	26
Table 5.1	Quadrotor UAV and Blade Flapping Model Simulation Parameters	106
Table 5.2	Baseline NLDI Simulation Virtual Controller Tuned Gains	109
Table 5.3	Standard Deviation of Turbulence Magnitude as a Function of Wind Speed and Turbulence Severity	112
Table 5.4	Turbulence Magnitude as a Percentage of the Wind Speed	113
Table 5.5	Performance Index Comparison under Nominal Conditions in Simulation	120
Table 5.6	Performance Index for Performance Comparison	123
Table 5.7	Wind Condition Sample Points for Monte Carlo Analysis.....	130
Table 5.8	Performance Metric Means for 500 Monte Carlo Runs	135
Table 5.9	Performance Metric Standard Deviation for 500 Monte Carlo Runs	135
Table 6.1	Main Properties of the DJI Flamewheel F330	137
Table 6.2	Baseline NLDI Flight Code Virtual Controller Tuned Gains.....	147
Table 7.1	Performance Index Comparison for Nominal Indoor Flight Tests	157
Table 7.2	Performance Index Comparison for Nominal Outdoor Flight Tests	164
Table 7.3	Performance Index Comparison for Low Wind Conditions.....	166
Table 7.4	Performance Index Comparison for Medium Wind Conditions.....	169
Table 7.5	Performance Index Comparison for High Wind Conditions	175

LIST OF FIGURES

Figure 2.1	General state-feedback control scheme	10
Figure 2.2	General cascade PID control scheme	12
Figure 2.3	Architecture of cascade backstepping control applied to quadrotor UAV (Huo, Huo, & Karimi, 2014)	16
Figure 2.4	General MRAC architecture with state predictor (Hovakimyan & Cao, 2010).....	20
Figure 2.5	General \mathcal{L}_1 state-feedback adaptive control scheme (Hovakimyan & Cao, 2010).....	22
Figure 2.6	General \mathcal{L}_1 output-feedback adaptive control scheme (Nshuti, 2017)	23
Figure 3.1	Body coordinate system with thrust and resultant torque convention for the motors.....	41
Figure 3.2	Moment arm definition for the quadrotor UAV system	44
Figure 3.3	Visualization of the ground velocity vector	46
Figure 3.4	Induced forces and moments due to blade flapping phenomena.....	50
Figure 4.1	General control law architecture of the NLDI inner loop applied to quadrotor system.	55
Figure 4.2	General control law architecture of the NLDI outer loop applied to quadrotor system.	55
Figure 4.3	Outer tracking loop design using cascade PID structure.....	61
Figure 4.4	Extended NLDI approach to compensate for wind effects.	65
Figure 4.5	General schematic for the ADALINE	73
Figure 4.6	General schematic for the RBFNN.....	75
Figure 4.7	General schematic for the SHLS-NN	79
Figure 4.8	Fast loop augmentation by SHLS-NN adaptive ANN	80
Figure 4.9	Augmenting \mathcal{L}_1 output-feedback adaptive controller	89
Figure 4.10	Fast loop augmentation by \mathcal{L}_1 output-feedback adaptive control.....	96
Figure 4.11	Bode plot for the low-pass filter designed for the simulation environment	99
Figure 4.12	Bode plot for the low-pass filter designed for the implementation environment.....	100
Figure 5.1	The DJI Flamewheel F330 simulation environment	105
Figure 5.2	DJI Flamewheel F330 simulation model.....	105
Figure 5.3	Sensor simulation model	107
Figure 5.4	Filtering and operations simulation model.....	107

Figure 5.5	Waypoint navigation simulation model.....	108
Figure 5.6	Control laws architecture simulation model.....	109
Figure 5.7	Adaptive ANN simulation model.....	110
Figure 5.8	Simulated wind environment.....	110
Figure 5.9	Wind speed magnitude time history with wind speed 6kts, turbulence severity 3 and wind direction 30deg from North.	113
Figure 5.10	Attitude angles tracking performance baseline NLDI under nominal conditions	118
Figure 5.11	Angular rates tracking performance baseline NLDI under nominal conditions	119
Figure 5.12	Translational and vertical tracking performance baseline NLDI under nominal conditions	119
Figure 5.13	Top view 3D simulated position hold performance comparison.....	122
Figure 5.14	3D simulated position hold performance comparison.....	122
Figure 5.15	Attitude angles time history comparison under wind disturbance conditions in simulation.....	124
Figure 5.16	Angular rates time history comparison under wind disturbance conditions in simulation	125
Figure 5.17	Baseline NLDI virtual control under wind disturbance conditions in simulation	125
Figure 5.18	Outputs of the adaptive ANN under wind disturbance conditions in simulation	127
Figure 5.19	NLDI with adaptive ANN augmentation virtual control under wind disturbance conditions in simulation.....	127
Figure 5.20	Outputs of the \mathcal{L}_1 adaptive controller for nominal indoor flight testing under wind disturbance conditions in simulation.....	128
Figure 5.21	NLDI with \mathcal{L}_1 adaptive controller augmentation virtual control under wind disturbance conditions in simulation.....	129
Figure 5.22	Controller comparison for attitude angles PM for wind 288 deg from North	132
Figure 5.23	Controller comparison for angular rates PM for wind 288 deg from North	133
Figure 5.24	Controller comparison for trajectory PM for wind 288 deg from North...	133
Figure 5.25	Controller comparison for PWM PM for wind 288 deg from North	134
Figure 5.26	Controller comparison for global PI for wind 288 deg from North	134
Figure 6.1	DJI Flamewheel F330 research testbed – front/top view	136

Figure 6.2	DJI Flamewheel F330 research testbed – side view.....	137
Figure 6.3	PX1 autopilot board (Px4 Dev Team, 2019).....	138
Figure 6.4	MPU6000 3-axis primary accelerometer/gyroscope (SparkFun Electronics, 2013).....	139
Figure 6.5	UBLOX LEA-6H GPS receiver module with digital compass (ArduPilot Dev Team, 2019).....	140
Figure 6.6	Lightware SF11/C laser (ACRONAME, 2019)	140
Figure 6.7	DX8 8CH transmitter and DSMX remote receiver (Horizon Hobby, 2019)	141
Figure 6.8	2212 920KV brushless motors (Amazon, 2019)	142
Figure 6.9	Simonk 30A ESC (Amazon, 2019)	142
Figure 6.10	B8x4.5MR-B4 propellers (Propellers, 2019)	143
Figure 6.11	HobbyStar 3000mAh 11.1V LiPo battery (RCJuice, 2019).....	143
Figure 6.12	Pixhawk Support Package library blocks (Pilot Engineering Group Mathworks, 2017)	144
Figure 6.13	S-Functions developed at the ADCL (Rivera K. , 2018)	145
Figure 6.14	General architecture of the implementation code used in the Flamewheel quadrotor	146
Figure 6.15	ULTIMETER 2100 wiring diagram showing all compatible sensors (Bros, 2017).....	148
Figure 6.16	ULTIMETER 2100 ground weather station.....	149
Figure 6.17	Sample WeatherText software wind measurement report.....	149
Figure 6.18	Location of Indoor Unmanned Systems Testing Facility inside the MicaPlex (Google, 2019)	151
Figure 6.19	Flight testing inside the Indoor Unmanned Systems Testing Facility.....	151
Figure 6.20	Artificial Turf Softball Field at Embry-Riddle Aeronautical University (Google, 2019)	152
Figure 7.1	Baseline NLDI roll and pitch rate virtual control for nominal indoor flight testing	158
Figure 7.2	Outputs of the adaptive ANN for nominal indoor flight testing	159
Figure 7.3	NLDI with adaptive ANN augmentation roll and pitch rate virtual control for nominal indoor flight testing	160
Figure 7.4	Outputs of the adaptive ANN showing instability for nominal indoor flight testing	161
Figure 7.5	NLDI with adaptive ANN augmentation roll and pitch rate virtual control showing instability for nominal indoor flight testing.....	161

Figure 7.6	Outputs of the \mathcal{L}_1 adaptive controller for nominal indoor flight testing....	162
Figure 7.7	NLDI with \mathcal{L}_1 adaptive controller augmentation roll and pitch rate virtual control for nominal indoor flight testing	163
Figure 7.8	3D trajectory comparison for nominal outdoor flight testing.....	165
Figure 7.9	3D trajectory comparison for low wind conditions	167
Figure 7.10	Outputs of the adaptive ANN for low wind conditions.....	168
Figure 7.11	Outputs of the \mathcal{L}_1 adaptive controller for low wind conditions	168
Figure 7.12	3D trajectory comparison for medium wind conditions.....	170
Figure 7.13	Roll and pitch angle response for baseline NLDI under medium wind conditions	171
Figure 7.14	Outputs of the adaptive ANN for medium wind conditions.....	172
Figure 7.15	Roll and pitch angle response for baseline NLDI with adaptive ANN augmentation under medium wind conditions	172
Figure 7.16	Outputs of the \mathcal{L}_1 adaptive controller for medium wind conditions.....	173
Figure 7.17	Roll and pitch angle response for baseline NLDI with \mathcal{L}_1 adaptive control augmentation under medium wind conditions	174
Figure 7.18	3D trajectory comparison for high wind conditions.....	175
Figure 7.19	Outputs of the adaptive ANN for high wind conditions.....	176
Figure 7.20	Outputs of the \mathcal{L}_1 adaptive controller for high wind conditions.....	177

SYMBOLS

$\tilde{e}_{TT}, \tilde{e}_\theta, \tilde{e}_\Omega, \tilde{e}_{PWM}$	Performance metrics
I_{xx}, I_{yy}, I_{zz}	Rigid body axial moment of inertia
I_{xz}, I_{xy}, I_{yz}	Products of inertia
K_β	Propeller stiffness
L_u, L_v, L_w	Scaling lengths for turbulence model
R_E^b	Earth to body reference frame transformation matrix
R_{prop}	Propeller radius
T_s	Sample time
w_i	Arbitrary weights in global PI
θ_0	Propeller collective pitch
θ_{tw}	Propeller linear blade twist
$\hat{\sigma}$	Estimated uncertainty in \mathcal{L}_1
ω_n	Natural frequency
h	Vertical offset of motors w.r.t. CG
Γ	Learning rate
F	External forces
I	Inertia matrix
K	Generic constant
L	Lateral offset of motors w.r.t. CG
M	External moments
T	Thrust
U	Virtual control
V	Linear velocities
V	ANN weights hidden to output layer
W	External wind
W	ANN weights input to hidden layer
X, Y, Z	X, Y and Z displacements in the inertial earth reference frame
b	ANN biases input to hidden layer
d	ANN biases hidden to output layer
e	Error signal
g	Gravity
k	Proportional gain
m	Mass
p, q, r	Body angular rates
r	Moment arm
u, v, w	Linear velocities in the body reference frame
\mathbb{I}	Identity matrix
Ω	Angular rates
β	Blade flapping angle
ζ	Damping ratio
λ	e-modification parameter
μ	Rotor advance ratio

σ	Hidden layer output of ANN
τ	Torque
ω	Angular vVelocity
ω_p	Angular velocity of the propeller
ϕ, θ, ψ	Euler angles

Operators and Notations:

$\ x\ _{\mathcal{L}_1}$	\mathcal{L}_1 norm of x
\vec{x}	Vector x
\dot{x}	Derivative of x
$\ x\ $	Euclidian norm of x
\hat{x}	Unit vector x
$ x $	Absolute Value of x
\ddot{x}	Second derivative of x
\hat{x}	Estimation of x
\tilde{x}	Error signal of x
x^2	x squared
x^T	Transpose of x
$x(0)$	Initial condition of x
$x(t)$	x as a function of time

Subscripts:

x_0	Initial condition of x
x_E	x defined in the inertial earth reference frame
x_T	x induced by thrust
x_X	x component in the earth X-direction
x_Y	x component in the earth Y-direction
x_Z	x component in the earth Z-direction
x_b	x defined in the body reference frame
x_d	x desired
x_e	Equilibrium point of x
x_g	x induced by gravity
x_i	i^{th} element of x
x_m	Model reference of x
x_w	x induced by wind
x_x	x component in the body x-direction
x_y	x component in the body y-direction
x_z	x component in the body z-direction
x_τ	x induced by torque

ABBREVIATIONS

6DOF	Six Degrees Of Freedom
ADALINE	Adaptive Linear Neuron
ADCL	Advanced Dynamics and Control Laboratory
AIS	Artificial Immune System
ANN	Artificial Neural Networks
DARPA	Defense Advanced Research Projects Agency
DCM	Direct Cosine Matrix
EMRAN	Extended Minimum Resource Allocating Network
ESC	Electronic Speed Controller
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IUSTF	Indoor Unmanned Systems Testing Facility
LED	Light Emitting Diode
LiPo	Lithium-ion Polymer
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MIMO	Multiple-Input Multiple-Output
MLP	Multi-Layer Perceptron
MRAC	Model Reference Adaptive Control
NLDI	NonLinear Dynamic Inversion
PI	Global Performance Index
PM	Performance Metric
PSD	Power Spectral Density
PWM	Pulse Width Modulation Signal
PX1	Pixhawk 1
RBFNN	Radial Basis Function Neural Network
RPM	Revolutions Per Minute
SD	Standard Deviation
SHLS-NN	Single Hidden Layer Sigmoidal Neural Network
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

ABSTRACT

Verberne, Johannes MSAE, Embry-Riddle Aeronautical University, May 2019.

Development of Robust Control Laws for Disturbance Rejection in Rotorcraft UAVs.

Inherent stability inside the flight envelope must be guaranteed in order to safely introduce private and commercial UAV systems into the national airspace. The rejection of unknown external wind disturbances offers a challenging task due to the limited available information about the unpredictable and turbulent characteristics of the wind. This thesis focuses on the design, development and implementation of robust control algorithms for disturbance rejection in rotorcraft UAVs. The main focus is the rejection of external disturbances caused by wind influences. Four control algorithms are developed in an effort to mitigate wind effects: baseline nonlinear dynamic inversion (NLDI), a wind rejection extension for the NLDI, NLDI with adaptive artificial neural networks (ANN) augmentation, and NLDI with \mathcal{L}_1 adaptive control augmentation. A simulation environment is applied to evaluate the performance of these control algorithms under external wind conditions using a Monte Carlo analysis. Outdoor flight test results are presented for the implementation of the baseline NLDI, NLDI augmented with adaptive ANN and NLDI augmented with \mathcal{L}_1 adaptive control algorithms in a DJI F330 Flamewheel quadrotor UAV system. A set of metrics is applied to compare and evaluate the overall performance of the developed control algorithms under external wind disturbances. The obtained results show that the extended NLDI exhibits undesired characteristics while the augmentation of the baseline NLDI control law with adaptive ANN and \mathcal{L}_1 output-feedback adaptive control improve the robustness of the translational and rotational dynamics of a rotorcraft UAV in the presence of wind disturbances.

1. Introduction

1.1. Background

The use of unmanned aerial vehicles (UAVs) for a multitude of applications has exponentially grown in recent years. Traditional UAV systems originate from military applications due to the vast reconnaissance and weapon deployment capabilities with minimized risk for friendly casualties. Examples of military UAV systems like the Firebee built by the Ryan Aeronautical Company, the AAI RQ-2 Pioneer, the General Atomics Predator and the Northrop Grumman Global Hawk have significantly influenced the outcome of US conflicts post World War II (Darack, 2011). The technological advancements and decrease in production cost in the twenty-first century resulted in UAV systems becoming available for commercial and private applications. Numerous examples show the positive influence these systems have had in disaster relief (Madrigal, 2011), law enforcement (Glaser, 2017), surveillance (McGivering, 2012), journalism (Kaufman & Somaiya, 2013), scientific research (Whitwam, 2016), global health (CBS News, 2018) and filmmaking (Lavrinc, 2012).

Besides all the benefits the introduction of UAV systems in the commercial and private sector have had to date, there are also limitations and potential threats to the public safety that are related to the continued use of these systems. The Federal Aviation Administration (FAA) predicts that non-commercial UAV ownership will rise from 1.1 million units in 2017 to 2.4 million units in 2022 in the United States alone (FAA, 2018). This results in examples, such as drones that fly in restricted areas (CBS News, 2014), the potential use of drones in terrorism (Gallagher, 2013) and the danger malfunctioning drones can have on large crowds (Weil, 2013), that will be seen more frequently in years

to come due to the inability to monitor and regulate the distribution of these systems in the private sector.

When it comes to the commercial UAV sector, the continued growth in autonomy and technology of UAV systems, together with the consumer's demand to make everyday life tasks simpler, will increase the application frequency of these systems. The projection is that the value of the commercial UAV systems market will rise from \$1 billion in 2017 to \$46 billion by 2026 (Cohn, Green, Langstaff, & Roller, 2017). Aside from further developing the current applications of UAV systems mentioned earlier, a large market left to be explored is personal air taxi service and package delivery. The latter is of particular interest to companies such as Amazon, UPS and Domino's Pizza (Desjardins, 2018), which are putting their efforts in developing the required technologies.

The aerospace industry's main concern with the exponential growth of private and commercial UAV usage is the safe introduction of these systems into the commercial airspace. The current focus of authorities is to safely separate airborne UAV systems from small and commercial aviation. The FAA presented a report in 2017 which showed that close encounters between UAV systems and commercial aviation grew from 874 in the period February through September 2015 to 1,274 in the same period in 2016 (FAA, 2017). A more current example showing the magnitude that these encounters can have is the December 2018 London Gatwick Airport shutdown due to reports of a singular drone flying near the airport, resulting in the stranding of 140,000 passengers and the affecting of 1,000 aircraft (Evans, 2018). Since the expectation is that the frequency of these encounters will rise further in following years, the call for an increase in regulation and safety of UAV systems is growing.

The prospected growth of the UAV private and commercial market and the subsequent push for the introduction of these systems into the airspace requires the managing of these units in a safe and controlled manner. This can be achieved by increasing the intelligence of individual UAV systems to allow for decision-making, path-planning and health-monitoring algorithms (Rivera, Moncayo, Verberne, & Festa, 2019) (Garcia D. F., 2017). However, a more efficient approach is it to manage the entire airspace using a generalized system. NASA is currently working on establishing an infrastructure to manage all these UAV systems in low-altitude (NASA, 2019). The proposed UAV traffic management system would provide safe and efficient path planning operations for all systems in the current airspace using information about traffic, weather and local terrain. The safe and efficient management of the airspace by a macro-managing traffic system cannot be guaranteed however if the UAV systems are not individually robust enough to guarantee the efficient tracking of navigational commands given by the traffic management system.

One side of assuring individual UAV systems are sufficiently robust is concerned with the security of the internal flight computer to prevent external security breaches. The hacking of UAV systems is a serious threat for public safety and attention is currently being placed on the cybersecurity of UAV systems (Krishna & Murphy, 2017).

Aside from guaranteeing a secure system to reject external cyber threats, the efficient operation of individual UAV systems also requires individual UAV systems to be inherently stable inside the flight envelope at all times. This includes the efficient rejection of internal systems failures and unknown external disturbances. Internal systems failures consisting of structural defects, sensor malfunction, actuator failures and software

glitches can be counteracted with robust (adaptive) control systems and redundant onboard systems. Examples can be found in (Moncayo, et al., 2013) where \mathcal{L}_1 adaptive control is applied to counteract actuator failures, (Perhinschi, et al., 2014) where bio-inspired artificial neural networks and an artificial immune system are applied to detect and identify actuator and aerodynamic failures, and in (Lopez, Dormido, Dormido, & Gomez, 2015) where H_∞ control is applied to counteract noisy sensor data.

The rejection of unknown external disturbances offers a more challenging task due to the limited available information about the disturbance. Arguably the most influential external disturbance to UAV systems is wind. The unpredictable and turbulent character of wind makes the topic of wind rejection one of the most researched in the UAV field. It is commonly known that the current infrastructure, especially in U.S. cities where buildings are placed in city blocks on lengthy streets, generates a wind channeling effect. This means that wind approaching a city is accelerated through the streets much like the wind in the test section of a wind tunnel. The introduction of UAV systems in these urban environments requires, therefore, robust control systems that can successfully reject wind effects so that these systems can operate in a safe and efficient manner at all times.

Fixed wing UAV systems often use external wind to their advantage to create favorable flight conditions to minimize energy. (Coulter, Moncayo, & Engblom, 2018) show how differential wind speed across altitudes can help sustain flight without propulsion for two tethered glider-type UAV systems, and (Langelaan, Alley, & Neidhoefer, 2011) show that if wind characteristics are known, the energy of a fixed wing UAV system can be minimized.

Unlike fixed wing systems, unmanned systems that rely on the production of vertical thrust by rotors to stay airborne are affected negatively by external wind due to the inability to use wind favorably. These systems are characterized by their ability to perform vertical take-off and landing (VTOL) combined with the capability to sustain flight while hovering in one place. External wind disturbance for these systems is solely seen as an external disturbance and needs to be rejected effectively to maximize the performance and reliability of these rotorcraft UAV systems.

The rejection of external wind effects by rotorcraft UAVs is generally attempted inside the control law architecture in one of two ways: effectively measuring or reconstructing the wind characteristics such that an estimate of the current wind can be used in the control law architecture, or by developing a robust control architecture that can efficiently reject the external disturbance with no need for wind information. The challenge in the first method is to initially obtain an accurate estimation or measurement of the wind characteristics and to then effectively use this information in the control architecture. The second method offers a more practical approach in the sense that no additional external sensors are needed to aid in measuring wind characteristics and less processing capabilities of the onboard computer are required to process wind measurements and/or estimators.

1.2. Research Objective

The main goal of this thesis is the design, development and implementation of robust control algorithms for disturbance rejection in rotorcraft UAVs. The main focus is the rejection of external wind effects. Four different control approaches are developed: a baseline NonLinear Dynamic Inversion (NLDI) controller, an analytical wind rejection

extension of the baseline NLDI control law, a baseline NLDI augmented with adaptive Artificial Neural Networks (ANN) and finally a baseline NLDI augmented with \mathcal{L}_1 output-feedback adaptive control. A simulation environment is applied to evaluate the performance of these four control algorithms under external wind conditions using a Monte Carlo analysis. Flight test results are presented for the implementation of the baseline NLDI, baseline NLDI augmented with adaptive ANN and the baseline NLDI augmented with \mathcal{L}_1 output-feedback adaptive control in a DJI F330 Flamewheel quadrotor UAV system. Flight testing is performed to evaluate the control algorithms in an indoor wind free environment where only ground and wall effects are present, and in an outdoor environment subjected to wind disturbance conditions. A set of metrics is applied to compare and evaluate the overall performance of the developed control algorithms and conclusions are drawn with respect to the effectiveness of each method in rejecting external wind effects.

1.3. Thesis Outline

Following the current chapter, Chapter 2 presents a literature review that includes a review of stabilizing linear, nonlinear, adaptive and intelligent control algorithms applicable to quadrotor UAVs, a general review of the stability criterion for nonlinear systems, an introduction to discrete time systems required for hardware-in-the-loop testing, and an overview of available wind estimation and rejection methods in UAV systems.

Chapter 3 presents the mathematical model and derives the equations of motion describing the translational and rotational motion of the dynamic quadrotor UAV system. This chapter also includes the mathematical wind model with induced forces and

moments acting on the quadrotor UAV used in the derivation of the extended NLDI and in the simulation environment.

Chapter 4 introduces the baseline NLDI and the derivation of the extended NLDI, followed by an introduction of the general architecture of the adaptive ANN and an overview of all studied adaptive ANN types. The adaptive ANN section is concluded with its application to augment the baseline NLDI and a brief stability analysis is shown to generate stability bounds on all the signals contained in the network. In a similar fashion, the general architecture of the \mathcal{L}_1 output-feedback adaptive controller together with the augmentation architecture is discussed, followed by a brief stability analysis. Chapter 4 concludes with the presentation of a discretized version of the \mathcal{L}_1 output-feedback adaptive controller required for implementation in the onboard computer of the DJI F330 Flamewheel quadcopter.

Chapter 5 introduces the simulation environment developed in MATLAB/Simulink. This includes a presentation of the simulation model, wind model and performance metric. Simulation results are shown for the four considered controllers in a Monte Carlo analysis applied to evaluate the performance of the controllers in a specified wind envelope.

Chapter 6 discusses the DJI Flamewheel F330 implementation testbed. An overview of the quadcopter components is presented together with a discussion of the software used to implement the developed control architectures. Next follows an introduction to the weather station used to characterize the wind in outdoor flight testing. Chapter 6 concludes with an introduction of the locations where flight testing was performed in order to evaluate the performance of the DJI Flamewheel in flight under

nominal and wind disturbance conditions.

Flight test results for the NLDI, NLDI augmented with ANN and NLDI augmented with \mathcal{L}_1 output-feedback adaptive control are presented in Chapter 7. Flight tests were performed in an indoor wind free environment and in an outdoor environment subjected to wind disturbance to evaluate the performance of the control techniques under nominal and disturbed conditions.

Using the results obtained in simulation and implementation, conclusions are drawn in Chapter 8 on the performance of each controller under external wind disturbance. This thesis is finalized with recommendations and future work summarized in Chapter 9.

The research effort presented in this thesis has resulted in various publications: (Rivera, Moncayo, Verberne, & Festa, 2019) and (Verberne, Betancur, Riverak, Coulter, & Moncayo, 2019) have been published, (Verberne & Moncayo, 2019) has been accepted for publication and (Verberne & Moncayo, 2019) is currently under review.

2. Literature Review

In general, UAV systems are underactuated systems. This is certainly true for quadrotor UAVs since four independent control inputs in the form of rotational speeds of the motors can be commanded while the system possesses six degrees of freedom (6DOF); three translational and three rotational. This results in a nonlinear, multiple-input multiple output (MIMO) system with a strong inherent coupling between translational and rotational dynamics (Wang, Man, Cao, Zheng, & Zhao, 2016). Stabilizing control with desired transient response is required to control the UAV throughout the flight envelope.

2.1. Stabilizing Control Architectures for Quadrotor UAVs

There are various approaches that can be taken when designing stable control laws for quadrotor UAVs. In general these approaches can be divided into linear control, nonlinear control, adaptive control and intelligent control. The application of control to UAV systems must operate in a closed-loop configuration such that information of the current performance can be used to drive the control input. Linear, nonlinear, adaptive and intelligent control approaches have been applied to control UAV systems in published literature. Although all these controllers can be tuned to enforce stable overall characteristics under ideal conditions, they distinguish themselves in the transient and steady state characteristics under (uncertain) internal and/or external disturbances. An overview of these methods applied to UAVs will follow next.

2.1.1. Linear control. The application of linear controllers to nonlinear systems such as UAVs will not result in a robust response due to their inherent design through linearization. The traditional pole placement theorem will be discussed first, which has significant drawbacks when applied to UAV systems. The inclusion of optimal control such as linear quadratic regulator (LQR) and H_∞ loop shaping improves the performance of the system.

Pole placement. For linear systems, the closed-loop stability can be altered by pole placement through state-feedback. Pole placement through state-feedback is considered the most fundamental form of control design. In regards to UAV systems, the application of pole placement requires current state information to allow for the stabilization of rotational and translational dynamics. These states can be sent back through a feedback loop with proportional gain K , as can be seen in Figure 2.1, to allow

for the altering of the closed-loop system dynamics. Since some states are not observable in the UAV system, e.g. the attitude angles, filtering and estimation is required to produce estimates for these states. This can be performed by, for example, Kalman filtering which will provide state estimates of unobservable states.

Instead of estimating unobservable states using state estimators, output feedback control can be applied to stabilize the dynamics of the system with the available outputs. Output feedback control has its limitations when applied to nonlinear, coupled systems like UAVs. Linearization of the nonlinear model results in a loss of nonlinear characteristics which are subsequently not compensated for by the linear output feedback controller. Also, the use of proportional feedback compensation has its limitations on the reachability of desired transient and steady-state stability features (Astrom & Murray, 2012).

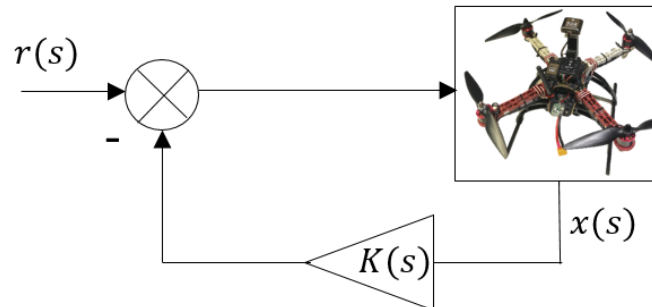


Figure 2.1. General state-feedback control scheme.

PID control. PID control offers a method to achieve both desired transient and steady-state characteristics of the dynamic system. PID compensation is the most popular linear control method applied to stabilize quadrotor UAVs due to the simplicity to tune the PID controller to achieve desired characteristics such as settling time, percent

overshoot and steady-state error (Garcia, Rubio, & Ortega, 2012). (Li & Li, 2011) and (Salih, Moghavvemi, Mohamed, & Sallom Gaeid, 2010) show that attitude and position can be stabilized by applying PID compensation on the error signal of the attitude angles and positions tracking.

In practice, traditional PID control applied to stabilize the rotational and translational dynamics is not very effective when large tracking errors exist which will degrade the transient performance of the system (Wang, Man, Cao, Zheng, & Zhao, 2016). One approach to counteract the unwanted features of traditional PID control to UAV systems is to apply PID control in cascade loops. The idea behind using a cascade architecture is to decouple the fast and slow dynamics to enforce desired transient response in both. The general cascade architecture can be seen in Figure 2.2. This cascade architecture can be applied to stabilize the fast dynamics for both the rotational and translational dynamics in the inner loop while stabilizing the slower dynamics in the outer loop. As seen in Figure 2.2, outputs of the outer loop act as inputs to the inner loop. Examples of cascade PID application to stabilize rotational dynamics are given in (Wang, Man, Cao, Zheng, & Zhao, 2016) where simulation results show that the cascade PID outperforms the traditional PID in the presence of disturbances, and in (Bo, Xin, Hui, & Ling, 2016) where PID and cascade PID flight test results are compared. A slightly different architecture of the cascade PID can be seen in (Cao & Lynch, 2016) where an inner-outer loop design is presented where translational dynamics are stabilized in the outer loop and rotational dynamics in the inner loop. This specific architecture promotes overall stable flight by coupling and stabilizing attitude and position states.

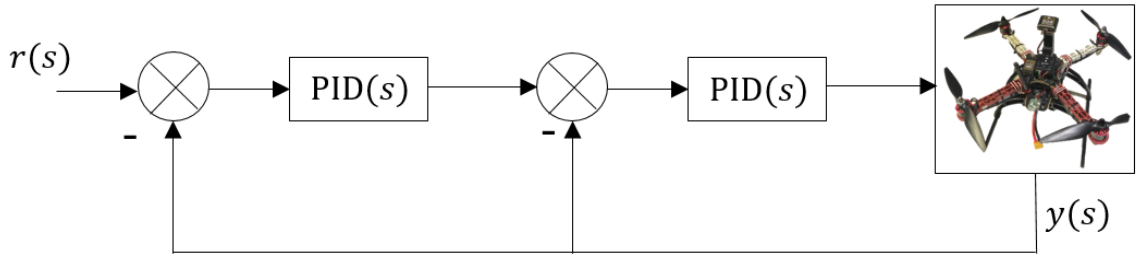


Figure 2.2. General cascade PID control scheme.

H_∞ control. Pole placement provides a method to determine the feedback gain required to achieve certain desired closed-loop stability characteristics. However, the obtained feedback gain might not be the most optimal solution to obtain the desired closed-loop characteristics. Originating from the 1960s, optimal control is concerned with finding adequate control inputs that will minimize a performance index penalizing system variables (Levine, 2011). One of the most popular optimal control applications is H_∞ loop shaping. H_∞ control is concerned with assuring overall stability for the closed-loop system by applying optimization in the frequency domain. This is achieved by finding a feedback gain that will minimize the maximum response for the closed-loop system in the frequency domain together with assuring closed-loop stability (Kwakernaak, 1993) (Zames, 1981). Minimizing the peak value of the frequency response can be achieved by applying the H_∞ norm to the closed-loop system (Levine, 2011):

$$\|G\|_\infty = \sup \sigma_{\max}[G(j\omega)] \quad (2.1)$$

where \sup represents the least upper bound of the closed-loop sensitivity function.

H_∞ control has been applied to stabilize quadrotor UAV systems. Examples can be found in (Chen & Huzmezan, 2003) where H_∞ loop shaping is applied for the

stabilization of rotational and translational dynamics of a quadcopter, and in (Falkenberg, Witt, Weltin, & Werner, 2012) where an aggressive H_∞ attitude controller is designed which shows superior results compared to a PID.

One drawback of the application of H_∞ control is the required extensive knowledge of control synthesis in the frequency domain as compared to more conventional methods.

Linear quadratic regulator. The linear quadratic regulator (LQR) problem is contained under the H_∞ control problem but offers implementation advantages, e.g. less information about the system is required to apply LQR and the computation of optimality is less involved. The LQR problem is concerned with finding the optimal state-feedback gain that will achieve guaranteed robustness while minimizing the energy spent (Levine, 2011). LQR requires a fully observable and controllable system. In the case that not all states are observable, like the general UAV system, an observer needs to be added to the LQR to provide state estimates. The LQR with observer is named the linear quadratic gaussian (LQG).

In short, the optimal feedback gain constructing the LQR controller can be calculated as:

$$K = -R^{-1}B^TP \quad (2.2)$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite tuning matrix penalizing control action for m inputs to the system, $B \in \mathbb{R}^{n \times m}$ is the input matrix in state-space representation for a system containing n states and $P \in \mathbb{R}^{n \times n}$ is the positive semidefinite solution of the algebraic Ricatti equation assuring that the input to the system, $u \in \mathbb{R}^m$, will minimize the cost function:

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.3)$$

where $Q \in \mathbb{R}^{n \times n}$ is a positive semi-definite tuning matrix penalizing transient state response.

Two examples of LQR control applied to quadrotor UAVs can be seen in (Dong, Fu, Yu, Zhang, & Al, 2015) where LQR control is effectively applied to control position and heading angle in flight, and in (Pan, Chen, Wang, Wu, & Cheng, 2018) where cascade LQR is applied to stabilize rotational and translational dynamics.

2.1.2. Nonlinear control. Section 2.1.1 discussed linear controller techniques designed to stabilize linear systems. The application of these controllers to nonlinear systems will not be ideal due to their linear synthesis. This section will emphasize nonlinear control design which is focused on altering the stability of nonlinear systems specifically. Gain scheduling is the first considered approach which focuses on counteracting an inherent drawback of control design through the linearization of nonlinear systems. Backstepping, sliding mode and nonlinear dynamic inversion (NLDI) control are all nonlinear methodologies concerned with effective and robust control tracking of a desired response using feedback. Backstepping and sliding mode control can be designed through the Lyapunov theorem while NLDI control relies on feedback linearization of the nonlinear system. Examples show that all the considered nonlinear control approaches yield a more robust system when applied to the nonlinear quadrotor UAV system in the presence of disturbances and uncertainties as compared to the linear control techniques.

Gain scheduling. Gain scheduling can be used in combination with the linear

controllers discussed in Section 2.1.1 as described in (Khalil, 2002). The main drawback in using a linearized model of a nonlinear dynamic system to tune and design stable control algorithms is the resulting loss of information due to linearization. The linearization of a nonlinear system only provides a relatively accurate representation of the nonlinear system around the linearization point. When the states of the system leave this linearization point, the linearized model diverges from the true nonlinear model. If the linearized model at a specific linearization point is used to design a stable controller, the controller performance will degrade once the states of the system leave the linearization point. Gain scheduling can provide a solution by linearizing the nonlinear system at various operation points inside the envelope and subsequently designing stable controllers at these points. The designed controllers can be merged together to allow for a smooth transition between controllers at different operating points (Harkegard, 2001). This will result in an overall more robust performance of the controller throughout the envelope.

An example of gain scheduling applied to quadrotor UAV systems can be seen in (Ataka, et al., 2013) where it is shown that a LQR gain scheduling architecture can be applied using a linearized quadcopter model to assure stable tracking of reference values by the quadrotor UAV.

Backstepping control. Backstepping control is a popular and effective approach to stabilize nonlinear systems. In a similar fashion as feedback linearization, which will be discussed later in this section, backstepping control applies (multiple) feedback loops to force the closed-loop system to exhibit favorable characteristics (Levine, 2011). Instead of minimizing a cost function, like is performed in H_∞ loop

shaping and LQR control, a virtual control input is generated using a Lyapunov approach such that the closed-loop system exhibits desired dynamic characteristics (Madani & Benallegue, 2006). When applied in a cascade architecture, emphasis must be placed on selecting an appropriate Lyapunov equation for each loop to ensure the output of the outer loop, which acts as a virtual control input to the inner loop, assures favorable transient and steady-state characteristics of the closed-loop system (Figure 2.3).

Examples of backstepping control applied to quadrotor UAV systems can be found in (Madani & Benallegue, 2006) where the backstepping control synthesis is described and applied for tracking translational and heading angle commands, and in (Huo, Huo, & Karimi, 2014) where backstepping control using a quaternion and integrator approach is shown to stabilize the attitude dynamics of a quadrotor in simulation.

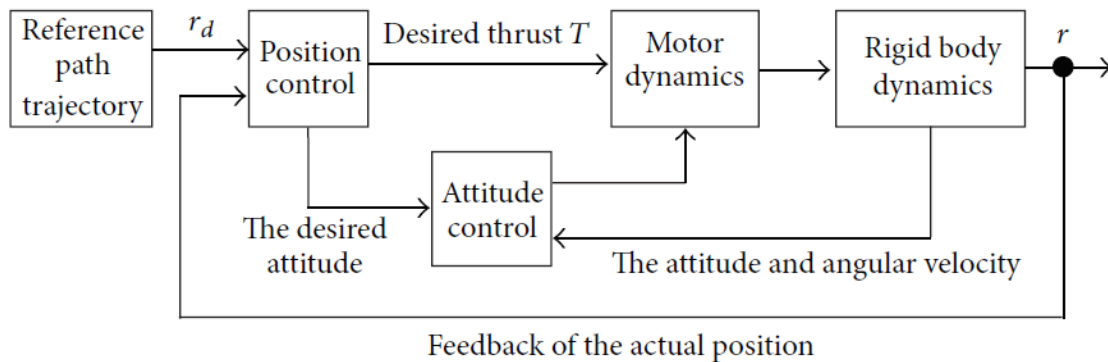


Figure 2.3. Architecture of cascade backstepping control applied to quadrotor UAV (Huo, Huo, & Karimi, 2014).

Sliding mode control. Sliding mode control is concerned with forcing the states of the system to track a desired pre-defined plane in the state space exhibiting

desired stability characteristics (Levine, 2011). The sliding mode controller will actively work towards keeping the state trajectory of the system on this pre-defined state plane. Sliding mode control design therefore consists of defining the desired state space plane exhibiting desired stability characteristics and defining a fast switching controller that will drive the system towards the desired response.

Similar to backstopping control, the sliding mode controller can be synthesized using a Lyapunov function. (Bouabdalla & Siegart, 2005) shows a sliding mode controller synthesized using a Lyapunov approach and subsequently applied to stabilize a quadrotor UAV. Another example can be found in (Xu & Ozguner, 2006) where a sliding mode controller is applied to stabilize the position and attitude of a quadrotor UAV. The sliding mode controller shows a robust response in the presence of parametric uncertainties.

Nonlinear dynamic inversion control. The general idea of NLDI control is to cancel the nonlinearities of a nonlinear system through feedback linearization. What remains is a linear system where classic control techniques such as pole placement and root locus can be applied to achieve desirable error dynamics. For NLDI to be applied effectively, a fairly accurate model of the nonlinear system dynamics must be available for application in the inversion loop.

A generic nonlinear system can be characterized as (Ito, Georgie, Valasek, & Ward, 2002):

$$\dot{\vec{x}} = \vec{f}(x) + g(x)\vec{u} \quad (2.4)$$

where $\vec{x} \in \mathbb{R}^n$ is the state vector, $\vec{u} \in \mathbb{R}^m$ are the inputs to the system, $\vec{f}(x) \in \mathbb{R}^n$

represents the nonlinear system dynamics and $g(x) \in \mathbb{R}^{n \times m}$ is the nonlinear input matrix. Given that $g(x)$ is invertible, feedback linearization can be applied to this nonlinear system by inverting Equation (2.4) as:

$$\bar{u}(x) = g^{-1}(x)[\bar{U}(x) - \bar{f}(x)] \quad (2.5)$$

where $\bar{U}(x) \in \mathbb{R}^n$ is a virtual controller which can be designed using classic control techniques to guarantee desirable stability dynamics. Most often the virtual control signal is generated using a simple linear controller which assures the desirable dynamics. When the input derived in Equation (2.5) is inserted in Equation (2.4), the closed-loop dynamics of the system approach the chosen desirable stability dynamics for a near perfect cancellation of nonlinear terms:

$$\dot{\bar{x}} \triangleq \bar{U}(x) \quad (2.6)$$

Unmodeled nonlinearities and uncertainties will be rejected by the pseudo controller driving the virtual control input $\bar{U}(x)$ in Equation (2.5) if the uncertainties are small. However, large uncertainties will result in large uncanceled dynamics after the application of feedback linearization which degrades the robustness of the controller.

The NLDI control law has been extensively used by researchers at West Virginia University to establish robust tracking of navigation commands by an YF-22 research testbed. Simulation results in (Moncayo, Perhinschi, Wilburn, Wilburn, & Karas, 2012) and flight test results in (Campa, et al., 2007) show desired tracking performance and fault tolerance capabilities of the NLDI control law applied to the YF-22 research testbed.

For quadrotor UAV systems, the NLDI control architecture has been implemented

in various research efforts. (Garcia D. F., 2017) shows the development of a NLDI control law for a quadrotor UAV system resulting in an improved robustness throughout the flight envelope. An application of this developed controller can be found in (Rivera, Moncayo, Verberne, & Festa, 2019) where implementation results show the robustness of the baseline NLDI controller with augmented adaptive control against failures and disturbances. In (Lee, Kim, & Sastry, 2009) feedback linearization is applied to stabilize the rotational and translational dynamics of a quadrotor UAV. Simulation results show that the sliding mode controller is more robust to external disturbances than feedback linearization.

2.1.3. Adaptive control. For the linear and nonlinear controllers discussed in Sections 2.1.1 and 2.1.2 respectively, the control parameters remain constant following the initial control design. This works if the system characteristics used to synthesize the control law are an accurate representation of the actual system. Unfortunately, all dynamic system models include modeling uncertainties and are vulnerable to internal and external disturbances. Adaptive control offers a solution to correct for these uncertainties and disturbances by changing its control parameters for accommodation.

Model reference adaptive control. Adaptive control originates from the 1950s when the need for robust autopilot systems was high for high performance aircraft with an extended flight envelope. Linear and nonlinear control algorithms could not guarantee the robustness throughout the extended flight envelope and so a new type of controller was required to guarantee a robust system for all expected flight conditions (Levine, 2011). One of the proposed methods to design such a new robust autopilot system was to base control output on a comparison between the current performance of

the aircraft with a desired reference model contained in a state predictor. This Model Reference Adaptive Control (MRAC) approach was first introduced by (Whitaker, Yamron, & Kezer, 1958) and a general architecture scheme can be seen in Figure 2.4.

One example of MRAC application can be found in (Dydek, Annaswamy, & Lavretsky, 2013) where an existing controller is augmented with MRAC to increase the robustness against parametric uncertainties in a quadrotor UAV in simulation and implementation. The results show that the MRAC augmentation increases the robustness of the system; however, there are some inherent shortcomings in the application of this controller.

As can be seen in Figure 2.4, at its core the MRAC compares state predictions with the current state of the actual system. The error between those two signals consists of modeling uncertainties and disturbances which are estimated using an adaptation law. The estimation of the modeling uncertainties and disturbances is fed back into the system for cancellation using a control law. The adaptive gain Γ can be increased to allow for fast adaption in the presence of uncertainties and disturbances. However, high frequency oscillations are prone to occur in the control signal for large errors between the state

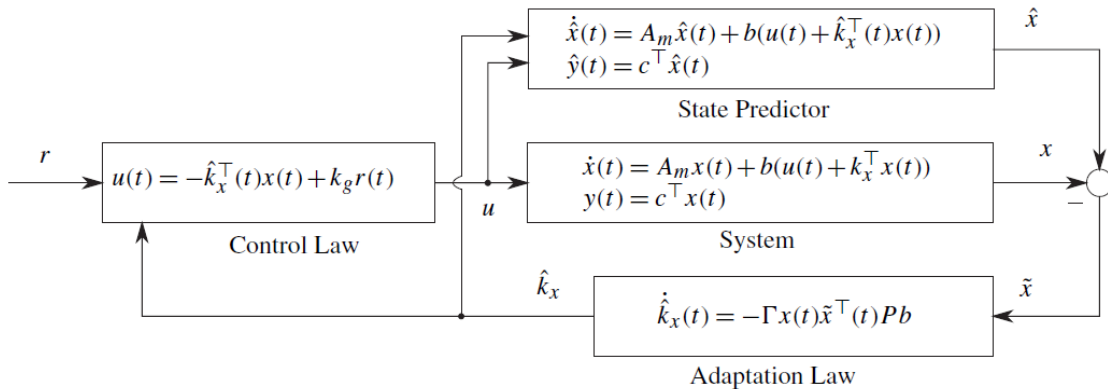


Figure 2.4. General MRAC architecture with state predictor (Hovakimyan & Cao, 2010).

predictor and system output leading to a decrease in stability of the system as is shown in (Cao, Patel, Reddy, & Hovakimyan, 2006). MRAC design therefore offers a tradeoff between fast adaptation and robustness.

\mathcal{L}_1 state-feedback adaptive control. The need for fast adaption and guaranteed robustness using model reference based control resulted in the evolution of the MRAC into the \mathcal{L}_1 adaptive control. \mathcal{L}_1 adaptive control was first introduced in (Cao & Hovakimyan, 2006a) and (Cao & Hovakimyan, 2006b) showing the desired decoupling between adaptation and robustness. Figure 2.5 shows the general \mathcal{L}_1 state-feedback adaptive control scheme for matched uncertainties. These uncertainties are considered to enter the system through the control channel, contained in the bandwidth of the control input. As can be seen, the \mathcal{L}_1 adaptive controller has a similar structure as the MRAC: state predictor, adaptation law and control law. The fast adaptation for modeling uncertainties and disturbances without sacrificing the robustness of the system is achieved by including a low-pass filter into the control design.

Results can be found in (Leman, Xargay, Dullerud, Hovakimyan, & Wendel, 2009) where \mathcal{L}_1 state-feedback adaptive control augmentation is applied to an X-48B aircraft model in simulation. Results show that the \mathcal{L}_1 augmented system is able to recover from failures. Another example can be found in (Moncayo, et al., 2013) where \mathcal{L}_1 state-feedback adaptive control augmentation is used to increase the tracking performance of the West Virginia University YF-22 simulation model under nominal and abnormal flight conditions.

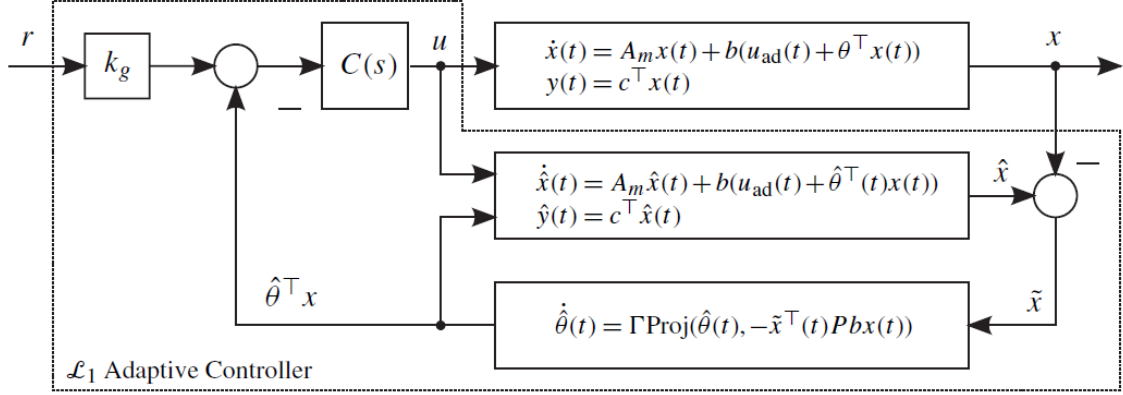


Figure 2.5. General \mathcal{L}_1 state-feedback adaptive control scheme (Hovakimyan & Cao, 2010).

\mathcal{L}_1 output-feedback adaptive control. Figure 2.5 shows that the application of \mathcal{L}_1 state-feedback adaptive control requires knowledge of all the states of the system. Since full state-feedback is often not feasible, a modified \mathcal{L}_1 adaptive control scheme can be applied which uses the available outputs of the system for feedback. Figure 2.6 shows a general \mathcal{L}_1 output-feedback adaptive control architecture for a stable reference model and strictly proper minimum phase filter as presented in (Hovakimyan & Cao, 2010). Notice that the \mathcal{L}_1 output-feedback adaptive controller has a similar architecture as the \mathcal{L}_1 state-feedback adaptive controller: an output predictor, an adaptation law and a control law. The \mathcal{L}_1 output-feedback architecture has no adaptive gain Γ available to increase the adaptation rate of the system. Fast adaptation and robustness of the system must be achieved by the design of the reference model and low-pass filter.

\mathcal{L}_1 output-feedback adaptive control has been applied in numerous research efforts. (Cao & Hovakimyan, 2009) shows the synthesis and application of \mathcal{L}_1 output-feedback adaptive control for the longitudinal control of a missile autopilot and

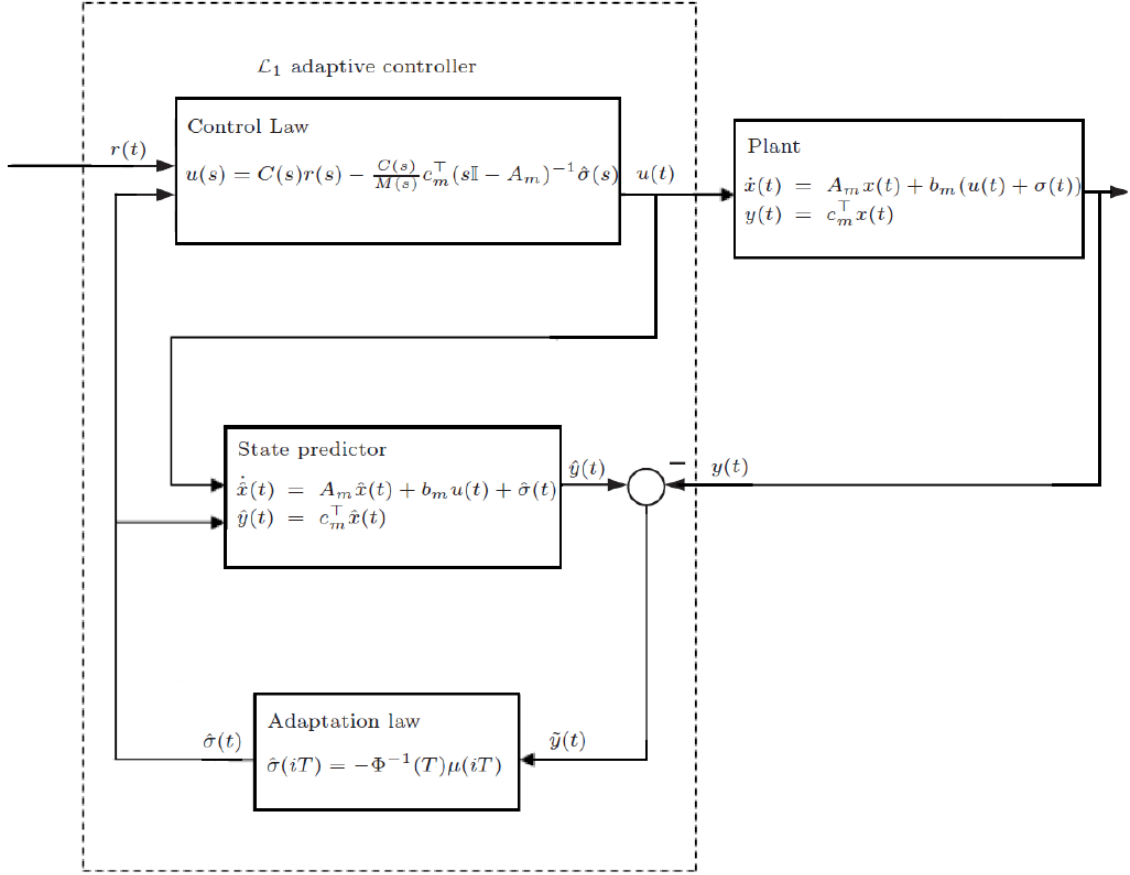


Figure 2.6. General \mathcal{L}_1 output-feedback adaptive control scheme (Nshuti, 2017).

(Geiser, Xargay, & Hovakimyan, 2011) shows how \mathcal{L}_1 output-feedback adaptive control can be used to augment a baseline NLDI controller for increased robustness in adverse conditions.

An example of \mathcal{L}_1 output-feedback adaptive control applied to UAV quadrotors can be found in (Suarez Fernandez, Dominguez, & Campoy, 2017) where the adaptive controller is compared for wind disturbance rejection capabilities to a LQR controller in simulation and with a PID controller in implementation by flying the quadrotor through a fan. Results show that the \mathcal{L}_1 outperforms both the LQR and PID.

2.1.4. Intelligent control. Control systems based on biological inspired

intelligent systems have become increasingly popular due to the potential to mimic human body functions that have been optimized through years of evolution (Levine, 2011). These systems can be characterized by their ability to learn from environmental information which will increase their robustness, accuracy and intelligence. Research efforts have been focusing on developing robust and trustworthy control applications of these algorithms. Arguably the three most researched biomimetic applications are fuzzy control, the artificial immune system (AIS) paradigm and artificial neural networks (ANN).

Fuzzy control. Fuzzy control attempts to mathematically model the instinctive heuristic understanding obtained through experiences that humans apply when controlling a dynamic system (Passino & Yurkovich, 1997). Where conventional control applied to dynamic systems focuses on system modeling and control through differential equations, fuzzy control focuses on establishing mathematical guidelines and decision-making algorithms. Fuzzy control has been applied to the control of quadrotor UAVs. An example can be seen in (Kim, 2018) where fuzzy control is applied for hovering control of a quadrotor UAV.

Artificial immune system paradigm. Another example of bio-inspired control is the application of the artificial immune system (AIS) metaphor for detection, identification, and evaluation of failures and uncertainties (Moncayo & Perhinschi, 2011). Like the human immune system, which successfully identifies and eliminates intruding antigens using a combination of antibodies and lymphocytes in a regulated feedback scheme, the AIS metaphor can be applied to identify abnormal flight conditions using mathematical antibodies trained with nominal flight data sets. An example can be

seen in (Garcia D. , et al., 2018) where the AIS paradigm is applied to recognize actuator failures and in (Rivera, Moncayo, Verberne, & Festa, 2019) where the AIS paradigm is applied to identify motor saturations in a quadrotor UAV in flight. The initial successful identification of abnormal flight conditions then yields the possibility to apply the AIS paradigm in adaptive control compensation. An example of the application of the AIS metaphor in adaptive control compensation can be seen in (Coulter, 2018) where a model reference AIS adaptive augmentation architecture is successfully applied to correct for undesired fuel slosh dynamics in spacecraft.

Artificial neural networks. Artificial neural networks (ANN) are mathematical models that simulate the working of the human nervous system (Al-Mahasneh, Anavatti, & Garatt, 2017). Originating from the 1940s, ANN have been applied in numerous applications due to their strong estimation, classification and prediction capabilities, with research of the last few decades focusing on their application to robust control systems (Table 2.1).

Like the human nervous system, ANN consist of a structure of interconnected neurons. In general, neurons are organized in layers with each individual neuron in each layer connected to all neurons in the previous and next layer. The first layer of neurons in the network make up the input layer while the last layer of neurons is denoted as output layer. The connections of neurons in between layers are weighted, with most neurons containing biases as well. The output of a neuron is generated according to an activation function which most commonly takes the shape of a sigmoidal function to create a bounded output between zero and one.

Table 2.1

The Application Development of ANN (Al-Mahasneh, Anavatti, & Garatt, 2017)

Before 1980	1980's	1990's	2000's	The last decade
Speech recognition, estimation, classification, data mining, robot planing and manipulator control.	System identification, process control, robots and robotic manipulators control, learning state space trajectories, pattern classification, forecasting.	control and system identification, pattern recognition, industrial process control, robotics, forecasting, aircraft control and development of EVNNs	adaptive and robust control, self evolving neuro-fuzzy control systems, fault diagnosis and co-operative learning	Self organising neural networks, hybrid fuzzy-neuro systems, machine learning, robotics, automotive, evolutionary control, UAVs Applications including: the non-linear neural control, adaptive control, robust control, model predictive control, neuro-PID control, neuro-evolutionary identification and neuro-evolutionary control.

The weights and biases contained in the network are available for tuning to assure the network behaves in a desirable fashion. The updating of the weights and biases is most effectively performed when applying supervised learning due to the use of an error signal in the tuning process, giving the network a direct measure of how well it is currently performing. The weights and biases are updated according to a backpropagation rule containing a gradient descent algorithm aimed towards varying the weights and biases in such a way to minimize the error. A stability proof is often tedious to show for ANN given the fact that the weights and biases of the ANN can be varied almost indefinitely resulting in an infinite amount of possible ANN schemes. Due to its inherent self-adaptive nature, ANN are notorious for their sometimes unpredictable behavior.

Many types of ANN have been developed and effectively applied. A few popular ANN types include the adaptive linear neuron (ADALINE), the single hidden layer sigmoidal neural network (SHLS-NN), the radial basis function neural network (RBFNN) with extended minimum resource allocating network (EMRAN) and sigma pi (AI-

Mahasneh, Anavatti, & Garatt, 2017). The ADALINE is a single layer network which does not use an activation function to obtain the output of individual neurons but instead uses a linear combination of the inputs. This results in a poor performance when applied to nonlinear systems. The SHLS-NN consists of neurons organized in an input layer, hidden layer(s) and output layer with sigmoidal activation functions. Unlike the ADALINE, the SHLS-NN is able to separate nonlinear data due to the increased complexity of the network. The RBFNN consists of a hidden layer containing neurons with a radial basis function. The output can therefore be denoted as a weighted summation of a finite number of radial basis functions. Besides the updating of the parameters of the radial basis functions by a backpropagation law, the addition of the ERMAN algorithm allows for the allocation and extraction of neurons in the hidden layer at locations where extra neurons are needed and where neurons do not significantly contribute to the output (Samy, Fan, & Perinpanayagam, 2010). Lastly, the outputs of the sigma pi network consist of a weighted combination of a set of basis functions (Rysdyk & Calise, 1998).

In the application of recognition and prediction, the neural network can be trained using nominal data featuring the desired characteristics. For example, (Perhinschi, et al., 2014) show how a RBFNN with EMRAN algorithm can be trained to mimic the nominal angular accelerations response of an aircraft, which can subsequently be used for failure identification in combination with the AIS paradigm. In this application, the learning would only be applied in the design process to train the ANN with the nominal data. When the network is considered properly tuned, the learning is discontinued and the network parameters remain constant.

Continuous learning can be applied for the application of ANN in a control law architecture yielding the adaptive ANN version. In this setup, the weights and biases contained in the network are continuously updated in an effort to minimize a given error signal. Adaptive ANN are commonly applied to augment existing controllers to increase the robustness of the system. The relative simplicity of control augmentation by ANN can be seen in (Sharma & Calise, 2005) where existing linear controllers are easily augmented with a SHLS-NN. The NLDI controller is often augmented by adaptive ANN to help estimate and correct for uncanceled nonlinearities in the feedback linearization process. Examples of this approach can be found in (Burken, Williams-Hayes, Kaneshige, & Stachowiak, 2006) where a sigma pi adaptive ANN is used to augment a NLDI control law to correct for inversion errors resulting from aerodynamic and control surface failures in a F-15 simulation. (Perhinschi, et al., 2005) show flight test results of a Gaussian radial basis function ANN, which is different from the RBFNN by the use of a Gaussian instead of a radial basis function, together with the EMRAN algorithm to augment a NLDI control law to compensate for inversion errors and changes in aircraft dynamics. Flight test results of the West Virginia University YF-22 research testbed show that the addition of adaptive ANN augmentation results in a more robust system.

2.2. Stability of Nonlinear Systems

A primary concern in any dynamic system is the concept of stability. Especially in aerospace applications, it is desired to confirm that the dynamic system is stable for all possible operating points inside the expected flight envelope. Conclusions on the stability of linear systems can be drawn by analyzing the eigenvalues and eigenvectors. A linear system is stable if all eigenvalues contain negative real components. Showing stability for

nonlinear systems is more challenging. One inefficient approach is to linearize the nonlinear system, which would only give information about the stability of the system at the linearization point. Different methods will need to be applied to prove stability for nonlinear systems.

2.2.1. General stability definitions. A general time-varying nonlinear system can be defined as:

$$\dot{\vec{x}} = f(\vec{x}(t), \vec{u}(t), t), \quad x(0) = x_0 \quad (2.7)$$

where $\vec{x} \in \mathfrak{R}^n$ is the state vector and $\vec{u} \in \mathfrak{R}^m$ are the inputs to the system. If the system remains at initial condition $(x_0, u_0) \forall t \geq t_o$ indefinitely for a constant control input, then $(x_0, u_0) = (x_e, u_e)$ is an equilibrium point of Equation (2.7). Nonlinear systems may contain more than one equilibrium point. An additional concern is the behavior of nonlinear systems when perturbed or initialized at a different point than the equilibrium point, or when moving from one equilibrium point to another inside the envelope. In general, three types of behaviors can be identified (Perez A. , 2016).

An equilibrium point x_e can be considered stable if and only if $\forall t \geq t_o$ and $\forall R > 0 \exists$ an initial condition $\|x(0)\| < r$ such that $\|x(t)\| < R$ where r and R are radii of spherical regions in the state space. Although the system does not return to the exact equilibrium point, the states of the system remain bounded in a finite region.

An asymptotically stable equilibrium point exhibits the same characteristics as the above defined stability definition when perturbed. Additionally, the states of the system return to the equilibrium point, that is initial condition $\|x(0)\| < r$ results in $x(t) \rightarrow x_e$ as $t \rightarrow \infty$.

An equilibrium point can be considered globally asymptotically stable when the states of the system converge to the equilibrium point for an infinite separation from the equilibrium point: initial condition $\|x(0)\| < \infty$ results in $x(t) \rightarrow x_e$ as $t \rightarrow \infty$.

2.2.2. Showing stability of equilibrium points. Now that stability concepts have been introduced, methods need to be applied to show that dynamic nonlinear systems converge to equilibrium points in a desired fashion.

Lyapunov's direct method. Arguably the most popular contemporary method to show stability for nonlinear systems is Lyapunov's direct method. Lyapunov's direct method applies the concept of total energy in the system and the change of this energy over time to draw conclusions on the overall stability of the system. If the dynamic system has an asymptotically stable equilibrium point, then the internal energy in the system will decay over time to its minimum value at the equilibrium point when the states are within spherical region $\|x(0)\| < r$ (Ogata, 1995). Lyapunov's direct method can be applied to show stability for both linear and nonlinear, time invariant and time variant systems. The selection of a valid candidate Lyapunov function representing the total energy contained in the system is often tedious and requires knowledge of the analyzed system. Lyapunov's direct method guarantees that the equilibrium points contained in the system are stable if there exists a continuously differentiable positive-definite function $V(x)$ such that Equation (2.8) results in a negative semidefinite result and asymptotically stable if Equation (2.8) results in a negative definite solution (Levine, 2011).

$$\dot{V}(x) = \frac{\partial V}{\partial x_i} \dot{x}_i \quad (2.8)$$

La Salle invariance principle. The La Salle invariance principle is a useful technique to show asymptotic stability of nonlinear systems. Equation (2.8) often yields negative semidefinite results, which cannot be used to infer asymptotic stability (Perez A. , 2016). The application of La Salle invariance principle allows for the drawing of conclusions with respect to the asymptotic stability of nonlinear systems with a negative semidefinite result obtained from Lyapunov's first method. This is performed by showing that the largest invariant set of points containing $\dot{V}(x) = 0$ is equal to the set containing the equilibrium points (Narendra & Annaswamy, 2005).

Barbalat's lemma. Barbalat's lemma forms an extension of the La Salle invariance principle and can be applied to Lyapunov candidate functions that result in a negative semidefinite uniformly continuous solution to Equation (2.8). Barbalat's lemma states that uniformly continuous functions are globally uniformly bounded and approach equilibrium as time approaches infinity and will thus result in asymptotic stability (Hovakimyan & Cao, 2010).

2.2.3. Stability of adaptive and intelligent control systems. The application of adaptive and intelligent controllers to increase the robustness of control systems is often complicated by the necessity for a stability proof assuring the desired behavior of the controller throughout the envelope. The methods discussed in Section 2.2.2 are usually applied to prove the stability of systems controlled by adaptive and intelligent controllers.

The usual approach for systems controlled by adaptive control techniques is to

express the system in closed-loop form and to subsequently show the boundedness of all signals in the closed-loop. (Cao & Hovakimyan, 2008) shows the application of an \mathcal{L}_1 output-feedback adaptive controller to control a generic plant. The stability of the closed-loop system is shown by proving the existence of upper bounds for the parameters in the system. Besides overall stability, it is inherently desirable for the adaptive controller to achieve the desired characteristics, which can be shown by analyzing the evolution of the closed-loop parameters over time.

Intelligent control often includes self-organizing or self-adapting elements for which it is important to show boundedness and convergence. In the case of self-adapting ANN, which contain inherent update rules for the parameters of the network, the boundedness of all parameters must be shown. In (Calise, Lee, & Sharma, 2001) the boundedness of all parameters inside a single hidden layer sigmoidal neural network (SHLS-NN) is shown by expressing the evolution of the parameters in the network as a Lyapunov function which can be shown to be negative definite proving the convergence of the network. Another example can be found in (Campa, Fravolini, Mammarella, & Napolitano, 2011) where a novel approach is shown to determine the stability of dynamic systems subject to output feedback direct adaptive control through the formulation of accurate relationships for the system's bounding sets.

2.3. Discrete Time Systems

The emergence of digital control systems to replace analog control systems has been catalyzed by the availability of low-cost digital computers and vast application possibilities (Ogata, 1995). Although dynamic systems can be considered continuous-time, controllers applied to these systems are inherently discrete due to the integration of

the controllers in a real-time sampled computer. This inherent discretization of the controller in a real-time environment must be taken into account in the controller synthesis.

In general, there are two approaches when it comes to designing discrete controllers (Jafarnejadsani, Lee, & Hovakimyan, 2017). The first approach is to express the open-loop of the continuous dynamic system as a discrete system and to subsequently synthesize the desired controller in the discrete Z-domain. The disadvantage of this method is that the discretization of the continuous-time model may not reflect the continuous-time system up to a desirable degree. The second approach is to synthesize the controller in continuous time and to emulate the continuous-time controller in discrete-time by assuring the sample time of the real-time environment is small. In essence, a continuous-time signal is a discrete-time system with a sample time that approaches zero. Sample time limitations are often induced by the available internal processing of the digital computer. The main concern in this approach though is whether the discretized controller will match the continuous-time designed control characteristics at the applied sample time.

The second approach described above will show poor performance for a large sample time. Due to the processing limitations on the sample time that onboard computers commonly pose, it is often attempted to try to evade the integration of high fidelity control systems on onboard computers. A popular strategy is to run controllers off-board on a ground station controlling the dynamic system through a telemetry connection, which is not limited to the internal processing capabilities of the onboard computer. This allows for fast processing off-board in combination with high frequency

telemetry connectivity. This approach is often taken when it comes to the application of reference model based adaptive control due to the straightforward design techniques available for the reference models and filters in the continuous-time domain. As a result, the controller can be run in a continuous-time environment with a very small sample time for the telemetry connection, mimicking the behavior of a continuous-time control system. (Suarez Fernandez, Dominguez, & Campoy, 2017) shows an example of this approach where continuous-time \mathcal{L}_1 adaptive control is applied to control a quadrotor UAV through telemetry commands sent from an off-board ground station. This approach allows for a 10,000Hz sampling frequency, which would be unattainable for the average onboard computer.

The stability of linear and nonlinear, time-variant and time-invariant discrete systems can be shown using Lyapunov's direct method discussed in Section 2.2.2. This method can be extended to allow for the inclusion of discrete time systems (Ogata, 1995).

2.4. UAV External Disturbance Rejection Methodologies

Many types of disturbances and uncertainties can degrade the overall performance of the UAV system. In general, disturbances and uncertainties can be categorized in either one of two classes: internal and external disturbances. Internal disturbances include those disturbances and uncertainties resulting from failures of hardware and software components including actuator failures, structural failures, on-board sensor malfunctions and software glitches. These undesired effects are commonly corrected for by applying the adaptive and intelligent control algorithms discussed in Sections 2.1.3 and 2.1.4. External disturbances include those effects caused by external influences acting on the system which are mostly weather related. Conventional UAV systems do not operate in

adverse weather conditions such as rain, snow and high wind. A realistic objective when it comes to external disturbances is therefore to reject low to medium wind effects acting on the UAV. This has been attempted in various ways.

2.4.1. Wind estimation methods. To attempt to correct for undesired wind effects, information about the current wind state conditions is required to be available. Effectively measuring or reconstructing the wind characteristics such that an estimation of the current wind velocity and/or acceleration can be used in the control laws architecture is a crucial initial step when it comes to successful external wind rejection. Onboard sensor data provided by sensors that are part of the standard autopilot unit can be used to obtain estimates of wind characteristics; in (Waslander & Wang, 2009) wind speed is estimated using accelerometer data and in (Langelaan, Alley, & Neidhoefer, 2011) a more extended approach is taken to obtain estimates for both wind speed and acceleration in simulation showing a relatively accurate estimation. (Sikkel, De Croon, De Wagter, & Chu, 2016) show a novel and relatively straightforward nonlinear observer to estimate the local wind components using accelerometer and GPS-velocity measurements in quadrotor UAV systems. Another example can be seen in (Tomic, Schmid, Lutz, Mathers, & Haddadin, 2016) where two methods are presented to estimate wind velocity. The first method applies an inverted model of the wind induced forces and moments for estimation while the second method obtains a wind velocity estimation based on the thrust of the motors. Both methods show an accurate estimation of the wind velocity in wind tunnel testing.

The addition of optical sensors to provide visual data can aid in the estimation of wind characteristics; (Abeywardena, Wang, Dissanayake, Waslander, & Kodagoda,

2014) uses a monocular camera to estimate wind velocity and in (Rodriguez, Andersen, Bradley, & Taylor, 2007) an optical flow sensor is applied to calculate wind velocity real-time on board a flying wing which can be used to estimate the crab angle.

Wind characteristics can be measured directly using additional sensors that require additional hardware which is often unfavorable; in (Yeo, Sydney, & Paley, 2015) a pressure probe flow measurement system is developed for a small quadrotor UAV to estimate the wind velocity. In (Palomaki, Rose, Van den Bossche, Sherman, & De Wekker, 2017), an anemometer attached to a quadrotor UAV is used to measure the wind velocity directly.

2.4.2. Wind rejection methods. Once wind characteristics are known, control law architectures can be developed in an attempt to correct for the external disturbance. One approach is to apply a correction into the control law architecture by modeling the aerodynamic effects caused by the external wind disturbance. (Hoffmann, Huang, Waslander, & Tomlin, 2007) presents corrections for two separate aerodynamic effects caused by external wind disturbance: thrust variation with free stream velocity, and induced roll and pitch moments resulting from a deflection of the thrust vector due to blade flapping. A continuation of this study in (Hoffmann, Huang, Waslander, & Tomlin, 2009) shows the implementation of the presented models in the control architecture of a quadrotor UAV and the increased robustness of the system to wind disturbance in both simulation and flight test. (Bannwarth, Chen, Stol, & MacDonald, 2016) shows the effective rejection of wind in flight test by a quadrotor UAV by using a disturbance accommodating control scheme which models the drag force by external wind to increase the position tracking capabilities. (Sydney, Smyth, & Paley, 2013) shows the estimation

of wind velocity using a recursive Bayesian filter and the inclusion of wind effects in the Z-domain in the inner and outer loop of a feedback linearization controller.

Besides modeling the aerodynamic effects caused by the external wind disturbance and including them in the system modeling and subsequent controller synthesis, another approach is to remove the requirement for an accurate system modeling completely. Section 2.1.2 showed that an accurate system modeling is required for a robust NLDI controller. The presence of wind induced disturbances and uncertainties result in an inaccurate system model which degrades the inversion performance of the NLDI. (Simplicio, Pavel, van Kampen, & Chu, 2013) introduces a modification of the classic NLDI control design by applying Taylor series expansion which does not require system model information but relies solely on sensor data for control. Flight test results of this so called incremental NLDI in a quadrotor UAV are shown in (Smeur, de Croon, & Chu, 2016). The incremental NLDI can be seen to considerably increase the robustness of the system when flying in and out the exhaust of a wind tunnel.

Lastly, the application of adaptive control algorithms offers a relatively straightforward approach in an effort to mitigate wind effects in UAV systems. This approach can be considered comparatively easy since there is no need for the modeling of wind effects in the system or control laws. An example of this approach can be found in (Escareno, Salazar, Romero, & Lozano, 2013) where a Lyapunov function based controller is used to correct for wind gust disturbances in a simulation environment of a quadrotor UAV. (Suarez Fernandez, Dominguez, & Campoy, 2017) shows the application of \mathcal{L}_1 output-feedback adaptive control in a quadrotor UAV system to reject

wind disturbances during high wind inspection applications. Results show that the \mathcal{L}_1 controller outperforms a LQR controller in simulation and a PID controller in flight testing when it comes to accurate tracking of reference values.

3. System Mathematical Modeling

This chapter presents mathematical models for the quadrotor UAV system and the complex wind environment. Rigid body kinematics and dynamics are applied to model the quadrotor UAV system and a mathematical approximation of the wind environment is derived. This is followed by an analysis of wind induced forces and moments assumed to act on the dynamic quadrotor UAV system.

3.1. Quadrotor UAV Model

3.1.1. Kinematics. In order to apply concepts of guidance and navigation to the quadrotor UAV system, methods are required to convert translational and rotational velocities from the vehicle body frame to the inertial reference frame. Kinematics of translation and rotation can be applied to achieve this conversion with the use of Euler angles.

Euler angles ϕ , θ and ψ provide an intuitive representation of a body in 3D space (Beard & McLain, 2012). The Euler angles can be used to convert from the inertial earth fixed reference frame to the body fixed reference frame using the rotation sequence ψ - θ - ϕ . The rotation matrix converting a vector from the inertial reference frame to the body reference frame through intermediate reference frames can be expressed as:

$$R_E^b = \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\phi s\theta - c\phi s\psi & c\psi c\phi + s\phi s\psi s\theta & c\theta s\phi \\ s\phi s\psi + c\phi c\psi s\theta & c\phi s\psi s\theta - c\psi s\phi & c\phi c\theta \end{bmatrix} \quad (3.1)$$

where the notation $\cos \alpha = c\alpha$ and $\sin \alpha = s\alpha$ is adopted. The attitude of the quadrotor

could be represented using quaternions but since the quadrotor will not approach high pitch angle values, Euler angles are preferred due to their more intuitive representation.

Linear velocities defined in the body reference frame can be converted to velocities in the inertial earth reference frame using the inverse of the rotation matrix defined in Equation (3.1) (Beard & McLain, 2012):

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta - c\phi s\psi & s\phi s\psi + c\phi c\psi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.2)$$

where $\vec{V}_b = [u \ v \ w]^T \in \mathbb{R}^3$ are the linear velocities expressed in the body reference frame and $\vec{V}_E = [\dot{X} \ \dot{Y} \ \dot{Z}]^T \in \mathbb{R}^3$ are the linear velocities expressed in the inertial earth reference frame.

Rotational velocities in the body reference frame can be converted to the inertial reference frame using a similar approach as for the translational velocities with the consideration of the rotation about the appropriate vector in each intermediate frame (Beard & McLain, 2012):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.3)$$

where $\vec{\Omega}_b = [p \ q \ r]^T \in \mathbb{R}^3$ are the angular accelerations expressed in the body reference frame and $\vec{\Omega}_E = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathbb{R}^3$ are the angular accelerations expressed in the earth reference frame.

3.1.2. Rigid body dynamics. Conservation of linear and angular momentum are applied to derive the rigid body dynamics of the quadrotor UAV system. A six

degrees of freedom (6DOF) rigid body under external forces and moments applied to the center of gravity in the body reference frame can be expressed in Newton-Euler form as (Napolitano, 2012):

$$m\dot{\vec{V}}_b + \vec{\Omega}_b \times m\vec{V}_b = \vec{F} \quad (3.4)$$

$$I\dot{\vec{\Omega}}_b + \vec{\Omega}_b \times I\vec{\Omega}_b = \vec{M} \quad (3.5)$$

where \vec{V}_b and $\vec{\Omega}_b$ are the linear and angular velocities expressed in the body reference frame, m is the mass of the quadrotor, $\vec{F} \in \mathbb{R}^3$ and $\vec{M} \in \mathbb{R}^3$ are the externally applied forces and moments expressed in the body reference frame, and $I \in \mathbb{R}^{3 \times 3}$ is the inertia matrix of the quadrotor where symmetry is assumed in all axes resulting in all off-diagonal values to be zero:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.6)$$

Equations (3.4) and (3.5) together with Equations (3.2) and (3.3) form the nonlinear equations of motion that describe the motion of the UAV quadrotor system over time.

The quadrotor system is modeled in a cross configuration. In this configuration all four motors contribute to generate angular rotations about the center of gravity, unlike the plus configuration where rotations about the x_b and y_b axes are produced by two independent sets of motors. Angular rotations and vertical displacement are directly produced by thrust and resultant torques that are used to control the quadrotor system. The cross configuration with the correct convention for the thrust forces and torques can

be seen in Figure 3.1.

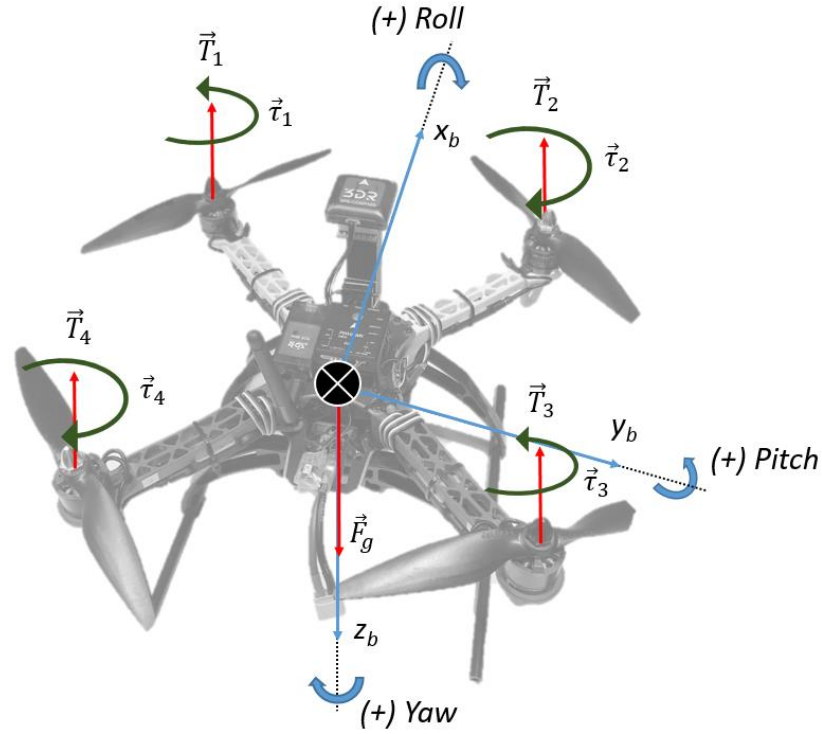


Figure 3.1. Body coordinate system with thrust and resultant torque convention for the motors.

External forces. Translational accelerations are a result of imbalanced external forces acting on the system, which can be represented as:

$$\vec{F} = \vec{F}_T + \vec{F}_g + \vec{F}_w \quad (3.7)$$

where $\vec{F}_T \in \mathbb{R}^3$ is the total thrust force required to sustain flight. The total thrust force is generated by the four individual motors which each produce one directional thrust force T_i :

$$\vec{F}_T = \begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=1}^4 T_i \end{bmatrix} \quad (3.8)$$

The thrust force generated by each of the four motors can be modeled as a function of the angular velocity of the propeller:

$$T_i = K_T \omega_i^2, \quad i = 1, 2, 3, 4 \quad (3.9)$$

where $K_T \in \mathfrak{R}$ is a constant describing the correlation between angular velocity and thrust, which is mainly dependent on the type of propeller, motor and electronic speed controller (ESC) used in the propulsion system of the quadrotor UAV. For the application of electronic motors requiring a pulse width modulation signal (PWM) as input, a characterization is required of angular velocity, thrust and torque as a function of PWM. This can be achieved through experimental testing to characterize the propulsion system (Garcia D. F., 2017).

$\vec{F}_g \in \mathfrak{R}^3$ defines the orientation of the gravitational forces with respect to the center of gravity of the quadrotor in the body reference frame:

$$\vec{F}_g = \begin{bmatrix} mg \sin \theta \\ -mg \cos \theta \sin \phi \\ -mg \cos \theta \cos \phi \end{bmatrix} \quad (3.10)$$

$\vec{F}_w \in \mathfrak{R}^3$ in Equation (3.7) models the wind induced external forces for which a detailed discussion is presented in Section 3.2.2.

External moments. The total external moments experienced by the quadrotor can be represented as:

$$\vec{M} = \vec{M}_T + \vec{M}_\tau + \vec{M}_w \quad (3.11)$$

where $\vec{M}_T \in \mathbb{R}^3$ models the moments produced by the thrust of the motors, $\vec{M}_\tau \in \mathbb{R}^3$ signifies the resultant torques acting on the quadrotor, and $\vec{M}_w \in \mathbb{R}^3$ consists of the moments caused by the wind drag forces acting on the propeller.

The moments produced by the thrust of the motors are a result of the differential between the individual thrust forces, which can be modeled for a UAV quadrotor with motors in the cross configuration following the convention in Figure 3.1 as:

$$\vec{M}_T = \begin{bmatrix} M_x \\ M_y \\ 0 \end{bmatrix} = \sum_{i=1}^4 (\vec{r}_i \times \vec{T}_i) = \begin{bmatrix} L(T_1 + T_4 - T_2 - T_3) \\ L(T_1 + T_2 - T_3 - T_4) \\ 0 \end{bmatrix} \quad (3.12)$$

where $\vec{r}_i \in \mathbb{R}^3$ is the moment arm between the center of gravity of the quadrotor and the thrust vector. The assumption is made that the quadrotor arms are perfectly modeled in the cross configuration such that the lateral distances between the center of gravity and the motors are constant and equal to L . All four motors are vertically offset by distance h . A visual representation for L can be seen in Figure 3.2.

$\vec{M}_\tau \in \mathbb{R}^3$ models the resultant torques allowing yawing motion of the quadrotor.

From Figure 3.1, the moments due to resultant torques can be represented as:

$$\vec{M}_\tau = \begin{bmatrix} 0 \\ 0 \\ M_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{bmatrix} \quad (3.13)$$

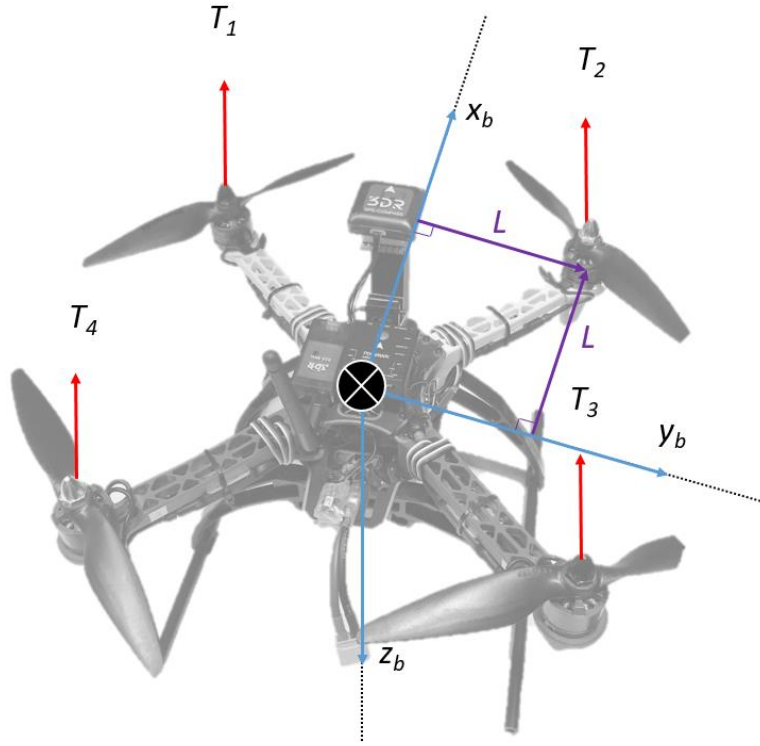


Figure 3.2. Moment arm definition for the quadrotor UAV system.

The resultant torque produced by each of the four motors can be modeled as a function of angular velocity of the propeller:

$$\tau_i = K_\tau \omega_i^2, \quad i = 1, 2, 3, 4 \quad (3.14)$$

where $K_\tau \in \mathfrak{R}$ is a constant describing the correlation between angular velocity and torque, which is also primarily dependent on the type of propeller, motor and electronic speed controller (ESC) used in the propulsion system of the quadrotor UAV. A characterization is required to model the motor produced torque as a function of angular velocity, which can be achieved through experimental testing (Garcia D. F., 2017).

Finally, $\vec{M}_w \in \mathfrak{R}^3$ in Equation (3.11) models the external wind induced moments, which will be further discussed in Section 3.2.3.

3.1.3. Control allocation. Equations (3.8), (3.12) and (3.13) can be combined and inversed to establish the thrust that must be commanded to each motor to achieve the desired moments and vertical force. The control allocation of the quadrotor UAV system with a cross configuration can be expressed as:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1/L & 1/L & -K_\tau^T \\ 1 & -1/L & -1/L & K_\tau^T \\ 1 & -1/L & 1/L & -K_\tau^T \\ 1 & 1/L & -1/L & K_\tau^T \end{bmatrix} \begin{bmatrix} F_{zd} \\ M_{xd} \\ M_{yd} \\ M_{zd} \end{bmatrix} \quad (3.15)$$

where F_{zd} , M_{xd} , M_{yd} and $M_{zd} \in \mathfrak{R}$ are the desired vertical force, rolling moment, pitching moment and yawing moment respectively generated by a control law desired for stability and navigation. $K_\tau^T \in \mathfrak{R}$ is an experimentally determined mapping factor to convert torque to thrust.

3.2. Wind Model

3.2.1. Modeling of the wind environment. A few assumptions are made with respect to the wind environment in order to provide a realistic model. Since quadrotor UAV systems are deployed in relatively low altitudes, a reasonable assumption is that the UAV will always operate in the lower atmospheric boundary layer. Therefore, the mean translational wind flow can be simplified to move in the horizontal directions only; parallel to the earth's surface with rotational wind effects neglected (Etele, 2006):

$$\vec{W}_E = \begin{bmatrix} W_X \\ W_Y \\ 0 \end{bmatrix} \quad (3.16)$$

3D atmospheric turbulence is added to the mean wind flow to account for the random wind variations in space and time:

$$\vec{W}_E = \begin{bmatrix} W_X + \Delta W_X \\ W_Y + \Delta W_Y \\ \Delta W_Z \end{bmatrix} \quad (3.17)$$

where ΔW_X , ΔW_Y and $\Delta W_Z \in \Re$ are the components of the added turbulence in the earth reference frame.

The wind vector in Equation (3.17) is assumed to act at the center of gravity (CG) of the quadrotor UAV. This allows for the inclusion of wind effects in the inertial earth reference frame to reflect the correct motion with respect to the ground:

$$\vec{V}_g = \vec{V}_E + \vec{W}_E = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} + \begin{bmatrix} W_X + \Delta W_X \\ W_Y + \Delta W_Y \\ \Delta W_Z \end{bmatrix} \quad (3.18)$$

where \vec{V}_E is the velocity of the quadrotor UAV in the inertial earth reference frame defined in Equation (3.2) and \vec{V}_g is the ground velocity vector which contains the proper velocity required for guidance and navigation for which a visualization can be seen in Figure 3.3.

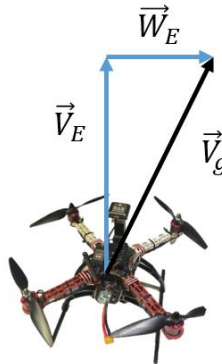


Figure 3.3. Visualization of the ground velocity vector.

In an effort to model the atmospheric turbulence, a few assumptions are made. The turbulence can be modeled as a stochastic process with the assumption that statistical properties are invariant in time and space, and with respect to rotations of coordinate systems. The effect of energy dissipation is neglected as well (Etele, 2006).

Turbulence can be modeled using the Dryden turbulence model, which is an approved mathematical model by the United States Department of Defense for the simulation of turbulence (Nshuti, 2017). The Dryden turbulence model applies power spectral density (PSD) functions deduced from measured aerodynamic data to describe the power distribution in frequency domain. For simulation purposes, a white noise based filter can be designed applying the power spectral density functions to model turbulence as will be discussed in Section 5.1.4.

The PSD functions according to the Dryden turbulence model for the air velocity components in the body reference frame are defined as (Etele, 2006):

$$\Phi_u(\Omega) = 2\sigma_u^2 L_u \frac{1}{1 + (\Omega L_u)^2} \quad (3.19)$$

$$\Phi_v(\Omega) = \sigma_v^2 L_v \frac{1 + 3(\Omega L_v)^2}{(1 + (\Omega L_v)^2)^2} \quad (3.20)$$

$$\Phi_w(\Omega) = \sigma_w^2 L_w \frac{1 + 3(\Omega L_w)^2}{(1 + (\Omega L_w)^2)^2} \quad (3.21)$$

where $\Omega \in \Re$ is the spatial frequency, σ_u , σ_v and $\sigma_w \in \Re$ are the standard deviations of the velocity components effectively acting as a turbulence severity, L_u , L_v and $L_w \in \Re$ are the scaling lengths for the power spectra.

3.2.2. Wind induced forces. $\vec{F}_w \in \mathfrak{R}^3$ in Equation (3.7) models the drag force caused by the relative wind vector defined in Equation (3.17) acting on the quadrotor. The drag force caused by the relative wind vector can be solved in the body reference frame as presented in (Bangura & Mahony, 2012) and (Omari, Hua, & Hamel, 2013):

$$\vec{F}_w = \vec{F}_{D_{aero}} + \vec{F}_{D_{turb}} + \sum_{i=1}^4 \left(\vec{F}_{d_{b.f.i}} + \vec{F}_{d_{i.d.i}} + \vec{F}_{d_{trans}i} + \vec{F}_{d_{prof}i} \right) \quad (3.22)$$

$\vec{F}_{d_{i.d.i}} \in \mathfrak{R}^3$ represents the induced drag term of the propellers. Induced drag inherently occurs due to the redirection of the airflow around the propellers to create lift and can be modeled to be proportional to the created lift. In the presence of a relative wind vector, the advancing blade sees a higher flow velocity than the retreating blade resulting in higher lift and therefore a higher induced drag term for the advancing blade in non-hover conditions opposing the relative wind direction. The induced drag of a spinning propeller in a relative wind field can be modeled to be directly proportional to the 2-D wind:

$$\vec{F}_{d_{i.d.i}} = K_{i.d.} \vec{W}_b \quad (3.23)$$

where $K_{i.d.} \in \mathfrak{R}$ is a constant determined through aerodynamic testing and \vec{W}_b is the wind vector expressed in the body coordinate system using the transformation matrix defined in Equation (3.1):

$$\vec{W}_b = R_E^b \vec{W}_E \quad (3.24)$$

$\vec{F}_{d_{trans}i} \in \mathfrak{R}^3$ represents the translational drag term. Translational drag occurs due

to the downward redirection of the flow going into the propeller. The airstream flowing into the propellers with the initial direction along the apparent wind vector is redirected downward by the spinning of the propellers. This change in direction creates a drag force along the relative wind direction. The translational drag term can be modeled as:

$$\vec{F}_{d_{transi}} = K_{trans} \vec{W}_b \quad (3.25)$$

where $K_{trans} \in \mathfrak{R}$ is a constant determined through aerodynamic testing.

The profile drag term, $\vec{F}_{d_{profi}} \in \mathfrak{R}^3$, models the skin friction and form drag the blades experience while they spin through the air. Form drag occurs due to the separation of the air flow while flowing over the spinning blade, while skin friction is inherent due to the viscous properties of air. Like the induced and translational drag terms, the profile drag term can be modeled proportional to the 2-D wind velocity vector represented in the body reference frame:

$$\vec{F}_{d_{profi}} = K_{prof} \vec{W}_b \quad (3.26)$$

where $K_{prof} \in \mathfrak{R}$ is a constant determined through aerodynamic testing.

$\vec{F}_{D_{aero}} \in \mathfrak{R}^3$ models the parasitic drag of the quadrotor frame with the propellers unattached. The largest contribution for this drag term is the form drag of the frame combined with the onboard equipment. The parasitic drag can be modeled as a second order function:

$$\vec{F}_{D_{aero}} = K_{aero} |\vec{W}_b| \vec{W}_b \quad (3.27)$$

where $K_{aero} \in \mathfrak{R}$ is constant which can be determined through wind tunnel testing.

$\vec{F}_{d_{b.f.i}} \in \mathfrak{R}^3$ models the aerodynamic forces induced by the blade flapping phenomena. Blade flapping occurs in non-rigid spinning blades during relative wind conditions. The advancing blade during the propeller rotation experiences a higher velocity and therefore higher induced lift than the retreating blade, which bends the propeller blade up and down respectively. This results in a tilt of the thrust vector away from the direction of wind, which induces a moment rotating the quadrotor away from the relative wind vector. An exaggerated representation of the blade flapping phenomena can be seen in Figure 3.4. The moment is induced since the motors in the quadrotor UAV are offset by distance h with respect to the CG as discussed in Section 3.1.2. Moment M_β is a concentrated moment caused by the bending of the rigid blades as will be analyzed in Section 3.2.3. Blade flapping also induces unbalanced forces at right angles from the relative wind vector. However, these forces are cancelled due to the quadcopter UAV cross configuration with counter-rotating motors. Only blade flapping forces parallel to

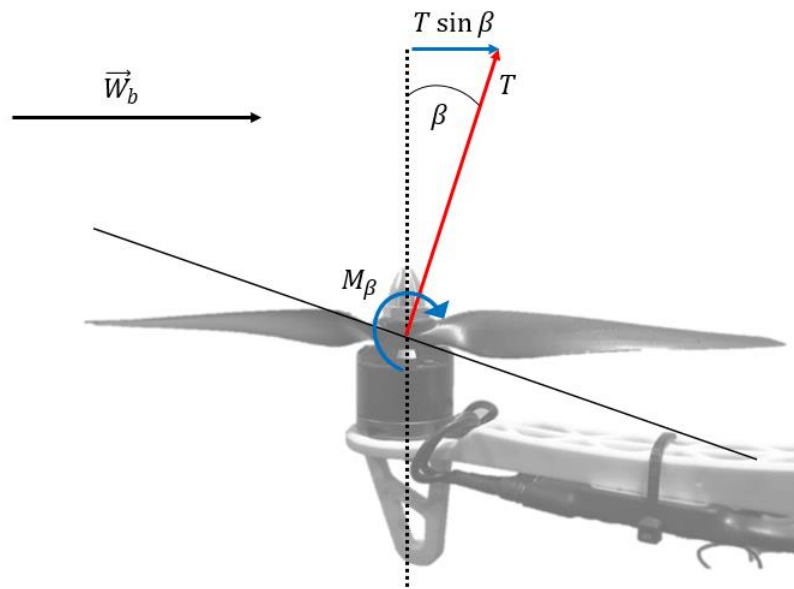


Figure 3.4. Induced forces and moments due to blade flapping phenomena.

the relative wind vector are therefore considered (Hoffmann, Huang, Waslander, & Tomlin, 2007). The aerodynamic force due to blade flapping can be modeled as (Huang, Hoffmann, Waslander, & Tomlin, 2009) :

$$\vec{F}_{d_{b.f.i}} = |\vec{F}_T| \sin \beta \hat{W}_b \quad (3.28)$$

where $\hat{W}_b \in \mathfrak{R}^3$ is the unit wind velocity vector, $|\vec{F}_T| \in \mathfrak{R}^3$ is the thrust force described in Equation (3.8) and $\beta \in \mathfrak{R}$ is the blade flapping angle, which can be approximated as outlined in (Leishman, 2000):

$$\beta = \frac{-\frac{8}{3}\mu(\theta_0 + \frac{3}{4}\theta_{tw})}{(1 - \frac{1}{2}\mu^2)} \quad (3.29)$$

where $\theta_0 \in \mathfrak{R}$ is the collective pitch of the propeller, $\theta_{tw} \in \mathfrak{R}$ is the linear blade twist of the propeller and $\mu \in \mathfrak{R}$ is the rotor advance ratio modeled as:

$$\mu = \frac{|\vec{W}_b|}{\omega_p R_{prop}} \quad (3.30)$$

where $\omega_p \in \mathfrak{R}$ is the angular velocity of the propeller and $R_{prop} \in \mathfrak{R}$ is the propeller radius.

The drag force due to turbulent airflow models the effects of the change in relative wind speed, which is experienced by the quadrotor as turbulence. This force can be represented as (Napolitano, 2012):

$$\vec{F}_{D_{turb}} = m\dot{\vec{W}}_b + \vec{\Omega}_b \times m\vec{W}_b \quad (3.31)$$

where $\dot{\vec{W}}_b$ is the wind acceleration vector obtained by taking the derivative of Equation (3.17) and transforming the wind acceleration vector from the inertial earth reference frame to the body frame:

$$\dot{\vec{W}}_b = R_E^b \dot{\vec{W}}_E \quad (3.32)$$

It is now possible to simplify Equation (3.22) with the assumption that all four propeller blades experience the same relative wind vector. Results obtained in (Bangura, 2017) are applied to remove the need for aerodynamic testing to approximate the constant magnitude drag coefficients:

$$\vec{F}_w = m\dot{\vec{W}}_b + \vec{\Omega}_b \times m\vec{W}_b + |\vec{F}_T| \sin \beta \hat{W}_b + 0.03\vec{W}_b \quad (3.33)$$

3.2.3. Wind induced moments. The moments due to wind drag force can be modeled by taking the cross product between the moment arm and the wind drag forces. Subsequently, the parasitic drag force and drag force due to turbulent airflow do not create a resultant moment. Therefore, the moments due to wind drag forces in Equations (3.23), (3.25), (3.26) and (3.28) can be modeled as:

$$\vec{M}_w = \vec{M}_\beta + \sum_{i=1}^4 (\vec{r}_i \times [|\vec{F}_T| \sin \beta \hat{W}_b + 0.03\vec{W}_b]) \quad (3.34)$$

where $\vec{r}_i \in \mathbb{R}^3$ represents the moment arm from the CG to one of four motors consisting of horizontal distances L and vertical offset distance h , $\vec{M}_\beta \in \mathbb{R}^3$ (Figure 3.4) is a concentrated moment due to the bending of the stiff rotor blades acting at the rotor shaft as adopted from (Huang, Hoffmann, Waslander, & Tomlin, 2009) and can be

approximated as:

$$\vec{M}_\beta = K_\beta \beta \hat{W}_b \quad (3.35)$$

where $K_\beta \in \Re$ is the stiffness of the propeller, $\beta \in \Re$ is the flapping angle as determined in Equation (3.29) and $\hat{W}_b \in \Re^3$ is the unit relative wind vector applied to allocate the appropriate moment addition to each direction.

4. Control Law Architectures

This chapter presents the stabilizing control law architectures applied to the quadrotor UAV. The nonlinear dynamic inversion (NLDI) is designed as a baseline control law for all architectures described in this thesis. In other words, three control law architectures modify or augment the baseline NLDI in an effort to mitigate wind effects. The proposed control laws include an analytical extension of the NLDI control law that implicitly uses wind effects, the baseline NLDI control law architecture with adaptive artificial neural networks (ANN) augmentation and finally the baseline NLDI control law architecture with \mathcal{L}_1 output-feedback adaptive control augmentation.

4.1. Baseline NLDI

Section 2.1 argued that the application of linear control to a nonlinear system will not result in a robust closed-loop system. Since the quadrotor UAV system is highly nonlinear and coupled as seen in Equations (3.4) and (3.5), nonlinear control must be applied to assure the nonlinear system behaves desirably. Section 2.1.2 discussed several nonlinear control approaches that can be applied to nonlinear systems. Gain scheduling could be applied to control the quadrotor UAV model but will require the linearization of the model at various points in the envelope. Although a grid of linearized models inside

the envelope can be used to generate a web of controllers, the robustness of the model will decay when the operation point moves away from the linearization points.

Backstepping and sliding mode control are other valid options for the control of the quadrotor UAV model but require tedious Lyapunov function design, while sliding mode control also requires knowledge of state plane design for the reference trajectories. The NLDI however is relatively easy to apply since a fairly accurate dynamic model is known as presented in Equations (3.4) and (3.5) which will result in a robust closed-loop quadrotor UAV system through the design of the virtual controllers.

The NLDI controller attempts to linearize the quadrotor UAV system through feedback linearization, which will cancel the nonlinearities in the system such that traditional linear controllers can be applied. Feedback linearization is achieved by inverting the equations of motion describing the quadrotor UAV system. The robustness and performance of the NLDI controller depend on the modeling accuracy of the equations of motion for the quadrotor UAV system.

The NLDI control architecture for the quadrotor UAV system consists of an outer and inner loop. The outer tracking loop is applied for navigation and guidance in autonomous flight while the inner loop, consisting of a slow and a fast loop, stabilizes the dynamics of the system. Figure 4.1 and Figure 4.2 show the control architectures of the NLDI inner and outer loop respectively applied to the UAV quadrotor system where the outputs of the outer loop provide the inputs to the inner loop (Garcia D. F., 2017). As it can be seen, inversion loops are connected in a cascade architecture where the 'Desirable Dynamics' blocks represent the linear controllers that can be applied to stabilize the feedback linearized system.

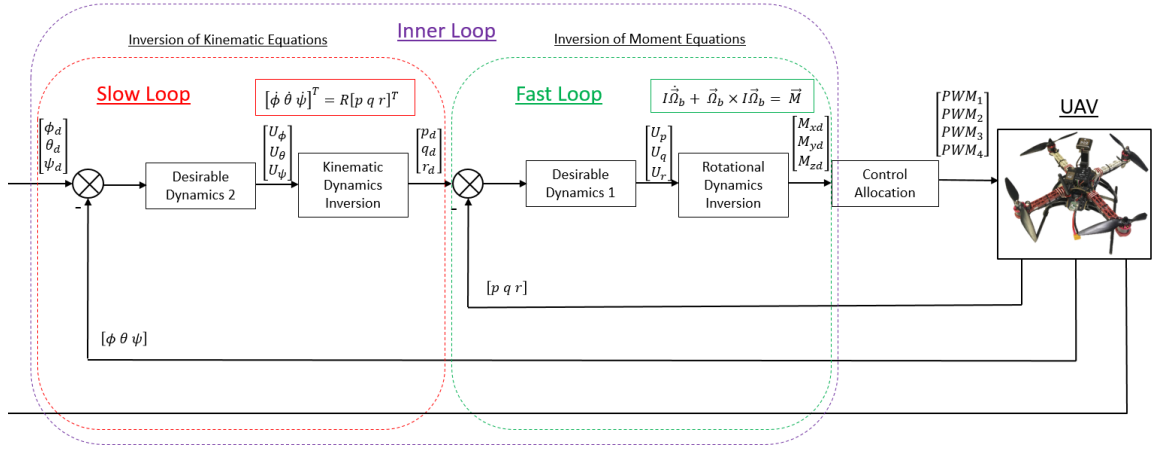


Figure 4.1. General control law architecture of the NLDI inner loop applied to quadrotor system.

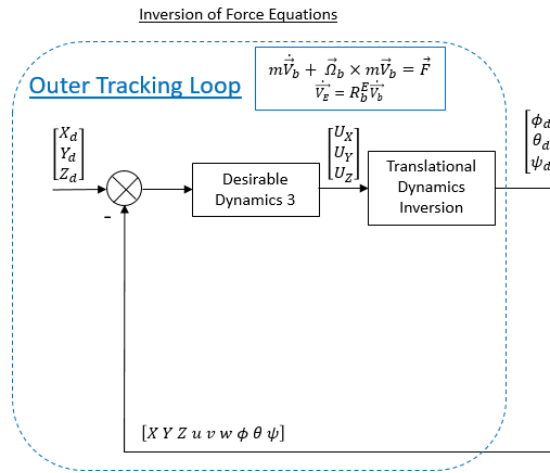


Figure 4.2. General control law architecture of the NLDI outer loop applied to quadrotor system.

4.1.1. Inner stability loop inversion. The objective of the inner loop inversion is to stabilize the rotational dynamics of the system. This is achieved by stabilizing the fast dynamics in the fast loop and the slow dynamics in the slow loop.

Fast loop. The fast loop inverts the conservation of angular moment equations

defined in Equation (3.5) to stabilize the fast dynamics of the quadrotor UAV system.

Inverting Equation (3.5) yields:

$$\begin{bmatrix} M_{xd} \\ M_{yd} \\ M_{zd} \end{bmatrix} = \begin{bmatrix} U_p I_{xx} \\ U_q I_{yy} \\ U_r I_{zz} \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})qr \\ (I_{xx} - I_{zz})pr \\ (I_{yy} - I_{xx})pq \end{bmatrix} - \Delta \bar{M} \quad (4.1)$$

where $\bar{M}_d = [M_{xd} \ M_{yd} \ M_{zd}]^T$ are the required moments to track the desired angular rates with desired dynamic stability characteristics modeled by the virtual controllers. $\Delta \bar{M}$ models external uncertainties and disturbances. The virtual controllers can be designed using methods discussed in Section 2.1.1. In the case that the virtual controllers consist of a simple proportional controller, they can be represented as:

$$\begin{bmatrix} U_p \\ U_q \\ U_r \end{bmatrix} = \begin{bmatrix} k_p(p_d - p) \\ k_q(q_d - q) \\ k_r(r_d - r) \end{bmatrix} \quad (4.2)$$

where k_p , k_q and k_r are proportional gains, and p_d , q_d and r_d are desired angular rates.

For a proper cancellation of nonlinear terms to take place, the inertias of the quadrotor UAV need to be known accurately and an internal gyroscope is required to supply angular rate measurements as can be seen in Equation (4.1). It is customary in the classic NLDI design to assume that a perfect inversion takes place such that uncertainties cancel in the feedback linearization process. The stabilizing linear virtual controllers applied to enforce the desired dynamic characteristics are able to compensate for the small uncertainties and disturbances represented in Equation (4.1) as $\Delta \bar{M}$. For an accurate cancellation of nonlinear terms, the closed-loop linearized system dynamics for the fast loop will be determined by the dynamics of the virtual controllers:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \cong \begin{bmatrix} k_p(p_d - p) \\ k_q(q_d - q) \\ k_r(r_d - r) \end{bmatrix} \quad (4.3)$$

The desired angular rates, $\bar{\Omega}_d = [p_d \ q_d \ r_d]^T$, can be generated by adding a slow loop in a cascade architecture.

Slow loop. The slow loop inverts the kinematics of rotation equations defined in Equation (3.3) to stabilize the slow dynamics of the quadrotor UAV. Inverting Equation (3.3) yields:

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}^{-1} \begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} \quad (4.4)$$

where $\bar{\Omega}_d = [p_d \ q_d \ r_d]^T$ are the required angular rates to track the desired attitude angles with desired dynamic stability characteristics modeled by the virtual controllers.

The virtual controllers can be modeled using a simple proportional control law:

$$\begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} = \begin{bmatrix} k_\phi(\phi_d - \phi) \\ k_\theta(\theta_d - \theta) \\ k_\psi(\psi_d - \psi) \end{bmatrix} \quad (4.5)$$

In order for the virtual controllers to enforce the desired dynamic stability characteristics on the slow dynamics of the quadrotor UAV, an accurate estimation of the attitude angles must be provided through, for example, Kalman filtering.

For an accurate cancellation of nonlinear terms, the closed-loop linearized system for the slow loop can be displayed as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \cong \begin{bmatrix} k_{\phi}(\phi_d - \phi) \\ k_{\theta}(\theta_d - \theta) \\ k_{\psi}(\psi_d - \psi) \end{bmatrix} \quad (4.6)$$

Virtual controller design. Now that the inversion for the fast and slow loop is achieved, attention must be placed in designing the virtual controllers in both loops to assure the desired response for the rotational dynamics. The closed-loop dynamics for the inner loop, which is the fast and slow loops combined, can be tuned with the method presented in (Wang, He, Zhang, & He, 2013). The closed-loop inner loop dynamics can be expressed as a system of differential equations:

$$\begin{aligned} \ddot{\phi} &= k_p(k_{\phi}(\phi_d - \phi) - \dot{\phi}) \cong k_p k_{\phi}(\phi_d - \phi) - k_p p \\ \ddot{\theta} &= k_q(k_{\theta}(\theta_d - \theta) - \dot{\theta}) \cong k_q k_{\theta}(\theta_d - \theta) - k_q q \\ \ddot{\psi} &= k_r(k_{\psi}(\psi_d - \psi) - \dot{\psi}) \cong k_r k_{\psi}(\psi_d - \psi) - k_r r \end{aligned} \quad (4.7)$$

The differential equations describing the closed-loop dynamics of the inner loop can be expressed in terms of the natural frequency and damping ratio such that gains k_p , k_q , k_r , k_{ϕ} , k_{θ} and $k_{\psi} \in \mathfrak{R}$ can be chosen to satisfy the desired transient dynamic characteristics:

$$\begin{aligned} \ddot{\phi} &\cong -\omega_{n\phi}^2(\phi_d - \phi) - 2\zeta_{\phi}\omega_{n\phi}p \\ \ddot{\theta} &\cong -\omega_{n\theta}^2(\theta_d - \theta) - 2\zeta_{\theta}\omega_{n\theta}q \\ \ddot{\psi} &\cong -\omega_{n\psi}^2(\psi_d - \psi) - 2\zeta_{\psi}\omega_{n\psi}r \end{aligned} \quad (4.8)$$

where ζ_ϕ , ζ_θ and $\zeta_\psi \in \Re$ are the damping ratios and $\omega_{n\phi}$, $\omega_{n\theta}$ and $\omega_{n\psi} \in \Re$ are the natural frequencies for the closed-loop rotational dynamics for each channel.

Combining Equations (4.7) and (4.8) produces:

$$\begin{aligned} k_\phi &= \frac{\omega_{n\phi}}{2\zeta_\phi} \\ k_\theta &= \frac{\omega_{n\theta}}{2\zeta_\theta} \\ k_\psi &= \frac{\omega_{n\psi}}{2\zeta_\psi} \end{aligned} \tag{4.9}$$

$$\begin{aligned} k_p &= 2\zeta_\phi\omega_{n\phi} \\ k_q &= 2\zeta_\theta\omega_{n\theta} \\ k_r &= 2\zeta_\psi\omega_{n\psi} \end{aligned} \tag{4.10}$$

4.1.2. Outer tracking loop inversion.

The desired attitude angles, $[\theta_d \ \phi_d \ \psi_d]^T$ required in Equation (4.6) can be directly linked to the remote controller inputs for pilot-in-the-loop flying. However, for autonomous operations the inner stability loop needs to be complemented with an outer tracking loop to provide the desired attitude angles based on waypoint navigation commands. In general, there are two approaches to design the outer tracking controller.

Cascade PID architecture. One approach to design the outer tracking control loop is to use a cascade PID structure. This approach does not allow for the cancellation of nonlinearities in the translational dynamics of the quadrotor UAV through feedback linearization. However, this method relaxes the modeling necessities of the dynamic system while still offering adequate robustness in the tracking control.

In the case that proportional linear controllers are applied, the desired attitude angles, $[\phi_d \ \theta_d \ \psi_d]^T$, and desired thrust force F_{zd} that act as inputs to the inner stability loop can be generated as:

$$\begin{aligned}\theta_d &= K_u(u_d - u) \\ \phi_d &= K_v(v_d - v) \\ \psi_d &= \psi_0 \\ F_{zd} &= K_w(w_d - w)\end{aligned}\tag{4.11}$$

where K_u, K_v and $K_w \in \Re$ are proportional gains, u_d, v_d and $w_d \in \Re$ are desired translational velocities expressed in the body reference frame, and u, v and $w \in \Re$ are the true translational velocities in the body reference frame. The desired yaw angle ψ_d for quadrotor UAV systems is often designed as a sample-and-hold control law where the take-off heading is maintained throughout the flight. The desired velocities in the body reference frame are generated by a second proportional controller:

$$\begin{bmatrix} u_d \\ v_d \\ w_d \end{bmatrix} = R_E^b \begin{bmatrix} K_{\dot{X}}(X_d - X) \\ K_{\dot{Y}}(Y_d - Y) \\ K_{\dot{Z}}(Z_d - Z) \end{bmatrix}\tag{4.12}$$

where $K_{\dot{X}}, K_{\dot{Y}}$ and $K_{\dot{Z}} \in \Re$ are proportional gains, X_d, Y_d and $Z_d \in \Re$ are the desired inertial earth reference frame locations, and X, Y and $Z \in \Re$ are the true inertial earth reference frame locations. Matrix R_E^b is defined in Equation (3.1). The velocity and position in the inertial earth reference frame of the quadrotor UAV are required to allow for error compensation. These can be supplied by a global positioning system (GPS). The

cascade PID configuration for the outer tracking loop shown in Figure 4.3 is applied to the implementation of the baseline NLDI control law in the quadrotor UAV for which flight test results will be shown in Chapter 7.

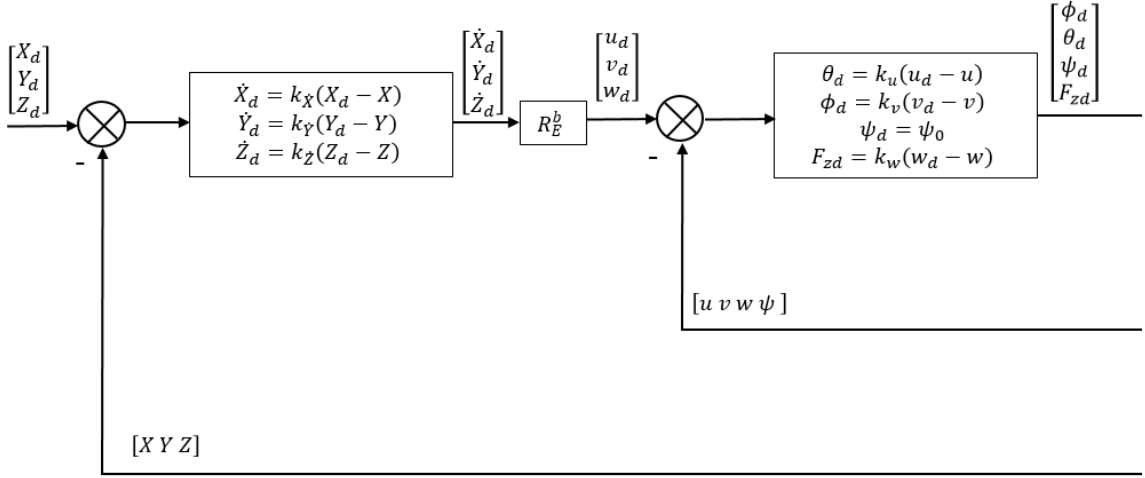


Figure 4.3. Outer tracking loop design using cascade PID structure.

In this architecture a linear controller is applied to a nonlinear system which does not guarantee optimal robustness. However, the application of a nonlinear controller to the outer tracking loop, such as feedback linearization, will increase the robustness of the closed-loop performance.

Feedback linearization architecture. A second approach to design the outer tracking control loop is to invert the force equations outlined in Equation (3.4) as is shown in Figure 4.2 (Ireland, Vargas, & Anderson, 2015). The application of feedback linearization allows for the cancellation of nonlinearities in the translational equations. The derivation starts with rewriting Equation (3.4) as:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = - \begin{bmatrix} -rv + qw \\ ru - pw \\ -qu + pv \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{F_T}{m} \end{bmatrix} + \Delta \vec{F} \quad (4.13)$$

where $\Delta \vec{F}$ models external disturbances and uncertainties which are assumed to cancel during the feedback linearization process in the classic NLDI design. Using the small angle assumption and neglecting the uncertainties and angular rate contributions, Equation (4.13) can be simplified to:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{F_T}{m} \end{bmatrix} \quad (4.14)$$

This result can be combined with the derivative of the kinematics of translation equations shown in Equation (3.2), which will allow for the conversion of accelerations in the body reference frame to the inertial earth reference frame:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta - c\phi s\psi & s\phi s\psi + c\phi c\psi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \quad (4.15)$$

Combining Equations (4.14) and (4.15), and solving for θ , ϕ and F_T will give the required inputs to feedback linearize the translational dynamics in the outer loop:

$$\theta_d = -\tan^{-1} \left(\frac{m(\cos \psi U_X + \sin \psi U_Y)}{F_T} \right) \quad (4.16)$$

$$\phi_d = -\sin^{-1}\left(\frac{m(\sin \psi U_X - \cos \psi U_Y)}{F_T}\right) \quad (4.17)$$

$$F_T = (U_Z + g)m \quad (4.18)$$

where small angle approximation is assumed in Equation (4.18).

U_X , U_Y and $U_Z \in \Re$ are virtual control inputs representing desired accelerations in the inertial earth reference frame. To generate the virtual control acceleration inputs from the desired waypoints, two cascade linear control loops can be applied similarly to the cascade PID architecture:

$$\begin{bmatrix} U_X \\ U_Y \\ U_Z \end{bmatrix} = \begin{bmatrix} k_{\dot{X}}(k_X(X_d - X) - \dot{X}) \\ k_{\dot{Y}}(k_Y(Y_d - Y) - \dot{Y}) \\ k_{\dot{Z}}(k_Z(Z_d - Z) - \dot{Z}) \end{bmatrix} \quad (4.19)$$

For an accurate cancellation of translational nonlinear terms, the closed-loop linearized system for the outer tracking loop can be displayed as:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \cong \begin{bmatrix} k_{\dot{X}}(k_X(X_d - X) - \dot{X}) \\ k_{\dot{Y}}(k_Y(Y_d - Y) - \dot{Y}) \\ k_{\dot{Z}}(k_Z(Z_d - Z) - \dot{Z}) \end{bmatrix} \quad (4.20)$$

where the proportional gains can be determined such that the closed-loop dynamics possess the desired transient characteristics:

$$\begin{aligned}
\ddot{X} &= k_{\dot{X}}k_X(X_d - X) - k_{\dot{X}}\dot{X} = -\omega_{nX}^2(X_d - X) - 2\zeta_X\omega_{nX}\dot{X} \\
\ddot{Y} &= k_{\dot{Y}}k_Y(Y_d - Y) - k_{\dot{Y}}\dot{Y} = -\omega_{nY}^2(Y_d - Y) - 2\zeta_Y\omega_{nY}\dot{Y} \\
\ddot{Z} &= k_{\dot{Z}}k_Z(Z_d - Z) - k_{\dot{Z}}\dot{Z} = -\omega_{nZ}^2(Z_d - Z) - 2\zeta_Z\omega_{nZ}\dot{Z}
\end{aligned} \tag{4.21}$$

$$\begin{aligned}
k_X &= \frac{\omega_{nX}}{2\zeta_X} \\
k_Y &= \frac{\omega_{nY}}{2\zeta_Y} \\
k_Z &= \frac{\omega_{nZ}}{2\zeta_Z}
\end{aligned} \tag{4.22}$$

$$\begin{aligned}
k_{\dot{X}} &= 2\zeta_X\omega_{nX} \\
k_{\dot{Y}} &= 2\zeta_Y\omega_{nY} \\
k_{\dot{Z}} &= 2\zeta_Z\omega_{nZ}
\end{aligned} \tag{4.23}$$

4.2. Extended NLDI with Wind Effects

This section shows the design of the robust analytical extension of the NLDI controller for wind rejection. The extended NLDI is an approach to analytically correct for wind induced effects on the dynamics of the quadroter UAV system. The derivation of this extension of the NLDI controller follows the derivation of the classic NLDI controller. However, the distinction is made in the inversion of the conservation of linear and angular momentum equations, Equations (3.4) and (3.5), performed in the outer navigation tracking loop and in the fast mode of the inner stability controller respectively.

Equations (4.1) and (4.13) show that it is customary in the derivation of the

classic NLDI control law to assume that the model of the dynamic system is accurate such that all external disturbances and uncertainties, $\Delta \vec{M}$ and $\Delta \vec{F}$, cancel in the feedback linearization process. However, this is not an accurate assumption due to the relative large influence of forces and moments caused by wind effects. Therefore, the feedback linearization does not completely cancel the forces and moments generated by wind, which degrades the performance of the controller. The new approach is therefore to include the modeled wind induced forces and moments discussed in Sections 3.2.2 and 3.2.3 in the derivation of the NLDI controller to allow for a more accurate cancellation of these nonlinearities in the feedback linearization, which will enhance the performance of the NLDI controller (Figure 4.4).

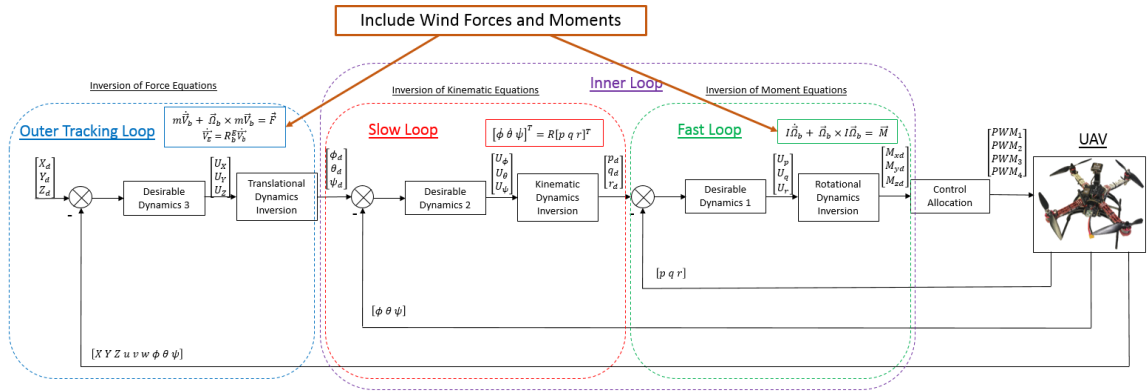


Figure 4.4. Extended NLDI approach to compensate for wind effects.

For the extended NLDI controller to be applied to the control architecture of the quadrotor UAV, a measurement or estimation of the wind velocity and wind acceleration is required. The overall robustness of the extended NLDI controller depends highly on the accuracy of the measured or estimated wind velocity and wind acceleration. Another concern is the modeling accuracy of the wind induced forces and moments presented in Sections 3.2.2 and 3.2.3. The assumption is that this model is sufficient to represent the

wind induced forces and moments accurately. If this is not the case, the robustness of the controller will be degraded. Note that for pilot-in-the-loop flying only the fast wind dynamics are compensated for in the extended inner stability loop inversion due to the exclusion of the outer tracking loop extension required for autonomous flight. These drawbacks will affect the overall improvement in robustness applying the extended NLDI as compared to the classic NLDI control law architecture.

4.2.1. Extended inner stability loop inversion. The derivation of the extended inner stability loop inversion starts with including the wind induced moments defined in Equation (3.34) into the conservation of angular momentum inversion shown in Equation (4.1) in the fast loop which yields:

$$\begin{bmatrix} M_{xd} \\ M_{yd} \\ M_{zd} \end{bmatrix} = \begin{bmatrix} U_p I_{xx} \\ U_q I_{yy} \\ U_r I_{zz} \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})qr \\ (I_{xx} - I_{zz})pr \\ (I_{yy} - I_{xx})pq \end{bmatrix} - \begin{bmatrix} M_{wx} \\ M_{wy} \\ M_{wz} \end{bmatrix} \quad (4.24)$$

where $[M_{wx} \ M_{wy} \ M_{wz}]^T$ are the moments produced by the wind drag forces, Equation (3.34), solved in the body reference frame. U_p , U_q and $U_r \in \mathfrak{R}$ are virtual control inputs representing desired angular transient characteristics and can be defined as in Equation (4.2).

For an accurate cancellation of all the wind induced moments in the inner tracking loop, the rotational dynamics will track the desired dynamics enforced by the virtual controllers:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \cong \begin{bmatrix} U_p \\ U_q \\ U_r \end{bmatrix} \quad (4.25)$$

By inspection, it can be seen that for excluding wind effects in the derivation of the extended NLDI inner loop control law, which means $\dot{W}_u = \dot{W}_v = \dot{W}_w = W_u = W_v = W_w = 0$, Equation (4.24) simplifies to Equation (4.1).

4.2.2. Extended outer tracking loop inversion. The derivation of the extended outer tracking loop inversion baseline NLDI control law starts with including the wind induced forces defined in Equation (3.33) in the conservation of linear momentum equation presented in Equation (4.13):

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = - \begin{bmatrix} -rv + qw \\ ru - pw \\ -qu + pv \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{F_T}{m} \end{bmatrix} + \begin{bmatrix} F_{wx} \\ F_{wy} \\ F_{wz} \end{bmatrix} \quad (4.26)$$

where $[F_{wx} \ F_{wy} \ F_{wz}]^T$ are the forces produced by the wind drag forces, Equation (3.33), solved in the body reference frame.

Equation (4.26) can be expanded as:

$$\begin{aligned} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} r(v - W_v) - q(w - W_w) \\ p(w - W_w) - r(u - W_u) \\ q(w - W_w) - p(u - W_u) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{F_T}{m} \end{bmatrix} \\ &+ \begin{bmatrix} \dot{W}_u \\ \dot{W}_v \\ \dot{W}_w \end{bmatrix} + \frac{1}{m} \begin{bmatrix} |\vec{F}_T| \sin \beta \hat{W}_u + 0.03W_u \\ |\vec{F}_T| \sin \beta \hat{W}_v + 0.03W_v \\ |\vec{F}_T| \sin \beta \hat{W}_w + 0.03W_w \end{bmatrix} \end{aligned} \quad (4.27)$$

Following the same derivation as for the classic NLDI derivation of the outer tracking loop, and combining Equation (4.27) with Equation (4.15), yields:

$$\begin{aligned}
\dot{W}_u + r(v - W_v) - q(w - W_w) + \frac{1}{m} |\vec{F}_T| \sin \beta \widehat{W}_u + \frac{0.03}{m} W_u \\
= c\psi c\theta \ddot{X} + s\psi c\theta \ddot{Y} - s\theta \ddot{Z}
\end{aligned} \tag{4.28}$$

$$\begin{aligned}
\dot{W}_v + p(w - W_w) - r(u - W_u) + \frac{1}{m} |\vec{F}_T| \sin \beta \widehat{W}_v + \frac{0.03}{m} W_v \\
= (c\psi s\phi s\theta - c\phi s\psi) \ddot{X} + (c\psi c\phi + s\phi s\psi s\theta) \ddot{Y} + (c\theta s\phi) \ddot{Z}
\end{aligned} \tag{4.29}$$

$$\begin{aligned}
\dot{W}_w + q(w - W_w) - p(w - W_w) + \frac{F_T}{m} + \frac{1}{m} |\vec{F}_T| \sin \beta \widehat{W}_w + \frac{0.03}{m} W_w \\
= (s\phi s\psi + c\phi c\psi s\theta) \ddot{X} + (c\phi s\psi s\theta - c\psi s\phi) \ddot{Y} + (c\phi c\theta) \ddot{Z}
\end{aligned} \tag{4.30}$$

Equations (4.28), (4.29) and (4.30) can be solved for θ , ϕ and F_T to obtained the desired pitch angle, roll angle and thrust force to effectively linearize the translational dynamics with wind effects in the outer tracking loop.

Equation (4.28) can be multiplied throughout with

$$\frac{1}{\sqrt{(c\psi \ddot{X} + s\psi \ddot{Y})^2 + (\ddot{Z})^2}} \tag{4.31}$$

which gives:

$$C = A \cos \theta + B \sin \theta \tag{4.32}$$

where

$$A = \frac{c\psi\ddot{X} + s\psi\ddot{Y}}{\sqrt{(c\psi\ddot{X} + s\psi\ddot{Y})^2 + (\ddot{Z})^2}}$$

$$B = \frac{-\ddot{Z}}{\sqrt{(c\psi\ddot{X} + s\psi\ddot{Y})^2 + (\ddot{Z})^2}} \quad (4.33)$$

$$C = \frac{\left(\dot{W}_u + r(v - W_v) - q(w - W_w) + \frac{1}{m}|\vec{F}_T| \sin \beta \hat{W}_u + \frac{0.03}{m}W_u\right)}{\sqrt{(c\psi\ddot{X} + s\psi\ddot{Y})^2 + (\ddot{Z})^2}}$$

It can be shown that:

$$A^2 + B^2 = 1 \quad (4.34)$$

which can be linked to the Pythagorean identity:

$$(\sin \alpha)^2 + (\cos \alpha)^2 = 1 \quad (4.35)$$

which means there is an unique angle α which satisfies:

$$\sin \alpha = A \quad (4.36)$$

$$\cos \alpha = B$$

Combining Equation (4.36) with Equation (4.32) gives:

$$C = \sin \alpha \cos \theta + \cos \alpha \sin \theta = \sin(\theta + \alpha) \quad (4.37)$$

Combining Equation (4.37) with Equation (4.33) and neglecting the angular velocity terms gives:

$$\sin(\theta + \alpha) = \frac{\dot{W}_u + \frac{1}{m} |\vec{F}_T| \sin \beta \hat{W}_u + \frac{0.03}{m} W_u}{\sqrt{(c\psi\ddot{X} + s\psi\ddot{Y})^2 + (\ddot{Z})^2}} \quad (4.38)$$

$$\theta + \alpha = \sin^{-1} \left(\frac{\dot{W}_u + \frac{1}{m} |\vec{F}_T| \sin \beta \hat{W}_u + \frac{0.03}{m} W_u}{\sqrt{(c\psi\ddot{X} + s\psi\ddot{Y})^2 + (\ddot{Z})^2}} \right) \quad (4.39)$$

Substituting $\alpha = \sin^{-1}(A)$ from Equation (4.36) in Equation (4.39) will give the desired pitch angle required to effectively linearize the outer tracking loop. A similar derivation can be shown to derive the required roll angle and desired thrust force.

The outputs of the outer tracking control loop that will effectively linearize the nonlinear translational dynamics are:

$$\theta_d = \sin^{-1} \left(\frac{\dot{W}_u + \frac{1}{m} |\vec{F}_T| \sin \beta \hat{W}_u + \frac{0.03}{m} W_u}{\sqrt{(c\psi U_X + s\psi U_Y)^2 + (U_Z)^2}} \right) \quad (4.40)$$

$$- \sin^{-1} \left(\frac{c\psi U_X + s\psi U_Y}{\sqrt{(c\psi U_X + s\psi U_Y)^2 + (U_Z)^2}} \right)$$

$$\phi_d = \sin^{-1} \left(\frac{-\dot{W}_v - \frac{1}{m} |\vec{F}_T| \sin \beta \hat{W}_v + \frac{0.03}{m} W_v}{\sqrt{(c\psi U_X + s\psi U_Y)^2 + (U_Z)^2}} \right) \quad (4.41)$$

$$+ \sin^{-1} \left(\frac{-s\psi U_X + c\psi U_Y}{\sqrt{(s\psi U_X - c\psi U_Y)^2 + (U_Z)^2}} \right)$$

$$\begin{aligned}
F_T = & -\dot{W}_w m + (s\psi s\phi + c\psi s\theta c\phi)mU_X \\
& + (-c\psi s\phi + s\psi s\theta c\phi)mU_Y + c\theta c\phi mU_Z
\end{aligned} \tag{4.42}$$

U_X , U_Y and $U_Z \in \mathfrak{R}$ are virtual control inputs representing desired accelerations in the inertial earth reference frame and can be defined as in Equation (4.19).

For an accurate cancellation of all the wind induced forces in the outer tracking loop, the translational dynamics will track the desired dynamics enforced by the virtual controllers:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \cong \begin{bmatrix} U_X \\ U_Y \\ U_Z \end{bmatrix} \tag{4.43}$$

It can be shown that the extended outer tracking loop inversion equations simplify to Equations (4.16), (4.17) and (4.18) for excluding wind effects in the derivation of the extended NLDI control law; that is, setting $\dot{W}_u = \dot{W}_v = \dot{W}_w = W_u = W_v = W_w = 0$ in Equations (4.40), (4.41) and (4.42).

4.3. Adaptive ANN Augmentation

The extended NLDI control law attempts to correct for wind induced effects analytically. As discussed in Section 4.2, the robustness of the extended NLDI relies on the accuracy of the UAV quadrotor model, the model of wind induced forces and moments and the measurement/estimation of wind velocity and acceleration. Uncertainties in the modeling, measurement and estimation will degrade the robustness of the extended NLDI controller. A second approach in an effort to mitigate wind effects on the UAV quadrotor system with a baseline NLDI control law is to apply intelligent

adaptive control. Adaptive ANN can be applied to estimate and correct for the uncertainties and disturbances in the system. By augmenting the baseline NLDI control law with adaptive ANN, the robustness of the baseline control law against uncertainties and disturbances can be improved.

4.3.1. Adaptive ANN selection analysis. Various types of ANN can be applied to dynamic systems for estimation, prediction and control as was discussed in Section 2.1.4. Three adaptive ANN structures were considered and compared for the application to a quadrotor UAV system.

Linear MIMO single layer neural network. The first and most fundamental artificial neural network that can be applied is the linear MIMO single layer artificial neural network (ADALINE) (Campa, Fravolini, & Napolitano, 2002). Compared to more complex types of ANN, the ADALINE does not use activation functions but instead produces the output using a linear combination of normalized inputs:

$$\vec{y}_{ADA} = W_{ADA}\vec{x}_{ADA} + \vec{\vartheta}_{ADA} \quad (4.44)$$

where $\vec{x}_{ADA} \in \mathbb{R}^n$ is a vector of normalized inputs, $\vec{y}_{ADA} \in \mathbb{R}^m$ is a vector containing the output of the ADALINE, $\vec{\vartheta}_{ADA} \in \mathbb{R}^m$ is a vector containing bias terms and $W_{ADA} \in \mathbb{R}^{n \times m}$ is a matrix containing the weights to be multiplied with the input signal. A general schematic for the ADALINE can be seen in Figure 4.5.

$\hat{W}_{ADA} = [\vec{\vartheta}_{ADA} \mid W_{ADA}]$ is a matrix containing the weights and biases updated according to a least squares rule which attempts to minimize the error between the output of the ADALINE, \vec{y}_{ADA} , and a reference value which can be represented in a discrete

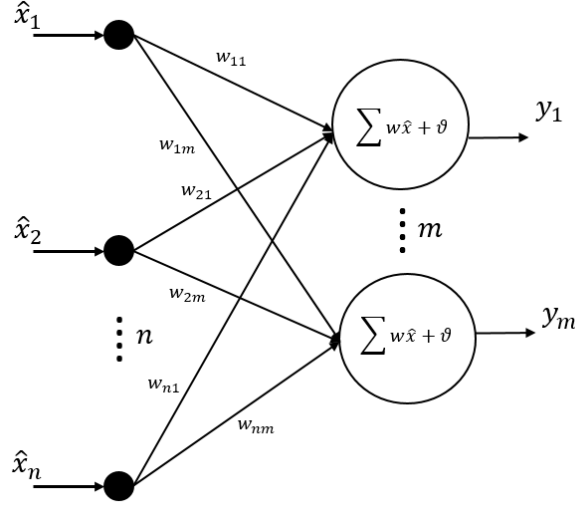


Figure 4.5. General schematic for the ADALINE.

format as (Rumelhart, Hinton, & Williams, 1986):

$$\hat{W}_{ADA}(k+1) = \hat{W}_{ADA}(k) - \Gamma_{ADA}(\vec{y}_{ADA}(k) - \vec{r}_{ADA}(k))\vec{x}_{ADA}(k) \quad (4.45)$$

where $\Gamma_{ADA} \in \Re$ is the learning rate, which is effectively an adaptive gain used to increase the adaptation rate of the network and $\vec{r}_{ADA} \in \Re^m$ are reference values to be compared to the output of the ADALINE. Since the ADALINE applies a linear combination of inputs, the network will not be robust in modeling nonlinear reference values. This makes the ADALINE not suitable for robust estimation of uncertainties and disturbances caused by wind effects in the quadrotor UAV system since these uncertainties and disturbances have a nonlinear character. A more complex artificial neural network is required to offer the estimation capacity desired.

Generalized MIMO radial basis function neural network. A more complex type of ANN is the MIMO radial basis function artificial neural network (RBFNN). Unlike the linear ADALINE, the RBFNN can be represented using the

weighted summation of a finite number of radial basis functions (Lu, Sundararajan, & Saratchandran, 2000) (Samy, Fan, & Perinpanayagam, 2010).

The i^{th} output of the RBFNN can be written as:

$$y_{RBFi} = \vartheta_{RBFi} + \sum_{v=1}^j w_{RBFv} \phi_v(\vec{x}_{RBF})_i \quad (4.46)$$

where $\vartheta_{RBFi} \in \mathfrak{R}$ is an estimated bias term and $w_{RBFv} \in \mathfrak{R}$ is an estimated weight multiplying the output of the v^{th} radial basis function, ϕ_v . The hidden layer consists of a total of j radial basis function hidden neurons. A general schematic for the RBFNN can be seen in Figure 4.6.

The radial basis function is often selected to be Gaussian:

$$\phi_v = e^{\left(\frac{-\|\vec{x} - \vec{\mu}_v\|^2}{\sigma_v^2} \right)} \quad (4.47)$$

where $\| \cdot \|$ represents the Euclidian norm between inputs \vec{x} and the center vector of the Gaussian function $\vec{\mu}$. σ represents the width of the Gaussian function. The network therefore consists of a web of neurons with Gaussian functions each having a different center and width, of which the outputs are weighted together to generate an estimation of a nonlinear function.

The network parameters are updated according to the gradient descent algorithm, which attempts to vary the weights, bias terms, widths of the Gaussian functions and centers of the Gaussian functions to minimize a given error signal. This can be represented in a discrete update law as (Samy, Fan, & Perinpanayagam, 2010):

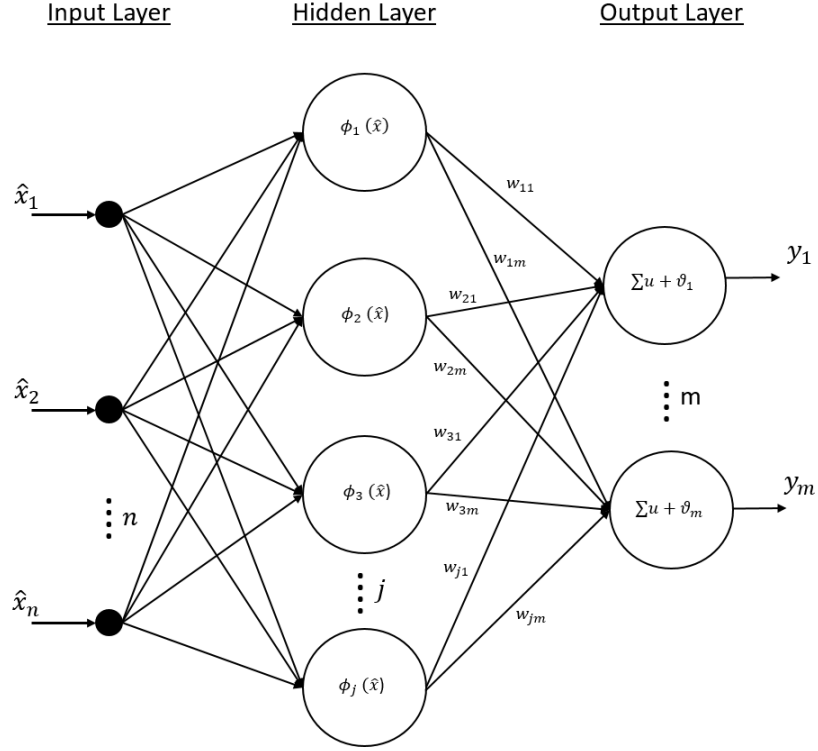


Figure 4.6. General schematic for the RBFNN.

$$\begin{aligned} \bar{\Delta}_{RBF}(k+1) = & \bar{\Delta}_{RBF}(k) - \Gamma_{RBF} \frac{\partial \bar{y}_{RBF}(k)}{\partial \bar{\Delta}_{RBF}(k)} (\bar{y}_{RBF}(k) - \bar{r}_{RBF}(k)) \\ & - \Gamma_{RBF} |\lambda| \bar{\Delta}_{RBF}(k) \end{aligned} \quad (4.48)$$

where $\bar{\Delta}_{RBF} \in \Re^{1 \times (j+m+2j)}$ is a vector containing the weights, bias terms, widths of the Gaussian functions and centers of the Gaussian functions to be updated by the update law, and $\lambda \in \Re$ is a stabilization factor to promote boundedness in the updating of the weights.

The RBFNN can be extended with the application of the extended minimal resource allocating network (EMRAN). The EMRAN attempts to optimize the size of the RBFNN by adding neurons in the hidden layer where they are most needed, and

removing unused neurons (Campa, Fravolini, & Napolitano, 2002). The RBFNN starts with no hidden neurons in the hidden layer. The EMRAN algorithm will add neurons when three criteria are met simultaneously (Samy, Fan, & Perinpanayagam, 2010):

$$e_k = \|\vec{y}_{RBFk} - \vec{r}_{RBFk}\| > C_1$$

$$e_{RMSk} = \sqrt{\sum_{q=k-(r-1)}^k \frac{e_q^2}{r}} > C_2 \quad (4.49)$$

$$d_k = \|\vec{\hat{x}}_k - \vec{\mu}_{nk}\| > C_3$$

where the first equation checks if the error between the reference values \vec{r}_{RBFk} and the outputs of the RBFNN \vec{y}_{RBFk} at time instant k is below a certain threshold C_1 . The second equation determines the root mean square of the last r time instants which is required to be below threshold C_2 . Finally, the third equation calculates the Euclidean norm between the inputs $\vec{\hat{x}}$ and the center vector of the Gaussian function $\vec{\mu}$ at time instant k . If all three errors are above the threshold, an extra neuron will be added in the hidden layer to increase the fidelity of the estimation.

The disadvantage of the application of the RBFNN with EMRAN to the quadrotor UAV system is the possible radical growth of the network with the addition of neurons in the hidden layer, which would have a considerable effect on the required computation. Initial simulation evaluations of the RBFNN with EMRAN showed the addition of extra neurons was required to increase the accuracy of the estimation of the network. The required accuracy was achieved at the cost of a considerable addition of hidden neurons, increasing the required computational power. This is a serious disadvantage for the

implementation of the RBFNN in an onboard computer of a quadrotor UAV system. The possible scenario would be that in flight the RBFNN starts adding neurons in the hidden layer to increase the fidelity of the estimation, which could shut down the flight computer when the maximum processing is surpassed. The RBFNN is therefore not the most optimal ANN to be applied to this quadrotor UAV scenario.

MIMO single hidden layer sigmoidal neural network. An ANN that does not change in size and can approximate nonlinear functions is the MIMO single hidden layer sigmoidal neural network (SHLS-NN) (Sharma & Calise, 2005) (Lewis, Yesildirek, & Liu, 1996). The network takes as inputs a set of normalized inputs:

$$\vec{\hat{x}} = [\hat{x}_1 \ \hat{x}_2 \ \dots \ \hat{x}_n]^T \quad (4.50)$$

These inputs are multiplied with individual weights specified on the connections between the neurons in the input layer and hidden layer. Note that each neuron in the hidden layer is connected with all the neurons in the input layer. The input to each neuron in the hidden layer consists of the sum of all the weighted inputs, to which a bias is added. The output from the hidden layer neurons is generated by feeding the sum of the weighted inputs with bias, which acts as an activation minimum, through a sigmoidal activation function, in an effort to create a bound on the output of the hidden layer neurons:

$$\sigma(u) = \frac{1}{1 + e^{-u}} \quad (4.51)$$

To generate the outputs of the neural network, each neuron in the output layer is connected to all neurons in the hidden layer through individual weighted connections. Again, a bias is added to the sum of the weighted inputs to the output layer. The general

schematic for the SHLS-NN can be seen in Figure 4.7 and the output of the network can be represented as:

$$y_q = \sum_{r=1}^j \left[v_{rq} \sigma \left(\sum_{s=1}^n w_{sr} \hat{x}_s + b_r \right) + d_q \right], \quad q = 1, \dots, k \quad (4.52)$$

which can be expressed in a shorter form as:

$$y_q = V^T \sigma(W^T \hat{x} + \hat{b}) + \hat{d} \quad (4.53)$$

where the biases and weights of the connections between the input layer and hidden layer, $\hat{W} = [\vec{b} \mid W] \in \Re^{j \times (n+1)}$, and between the hidden layer and output layer, $\hat{V} = [\vec{d} \mid V] \in \Re^{k \times (j+1)}$, are being updated by a backpropagation law (Sharma & Calise, 2005):

$$\dot{\hat{W}} = -\Gamma_1 [\vec{\hat{x}} e \hat{V}^T \hat{\sigma}' + \lambda_1 |e| \hat{W}] \quad (4.54)$$

$$\dot{\hat{V}} = -\Gamma_2 [(\hat{\sigma} - \hat{\sigma}' \hat{W}^T \vec{\hat{x}}) e + \lambda_2 |e| \hat{V}]$$

where $\Gamma \in \Re$ is the learning rate of the update law, $e \in \Re$ is the error signal that drives the updating of the weights and biases, $\hat{\sigma} \in \Re^k$ is the output of the hidden layer, $\hat{\sigma}'$ is the Jacobian with respect to the weights and biases, and finally $\lambda \in \Re$ is the e-modification parameter that counteracts uncontrolled parameter growth commonly found in adaptive control theory.

The SHLS-NN is chosen as the desired ANN for application to the quadrotor UAV system to mitigate wind effects since the SHLS-NN approximates nonlinear terms and does not change size while doing so.

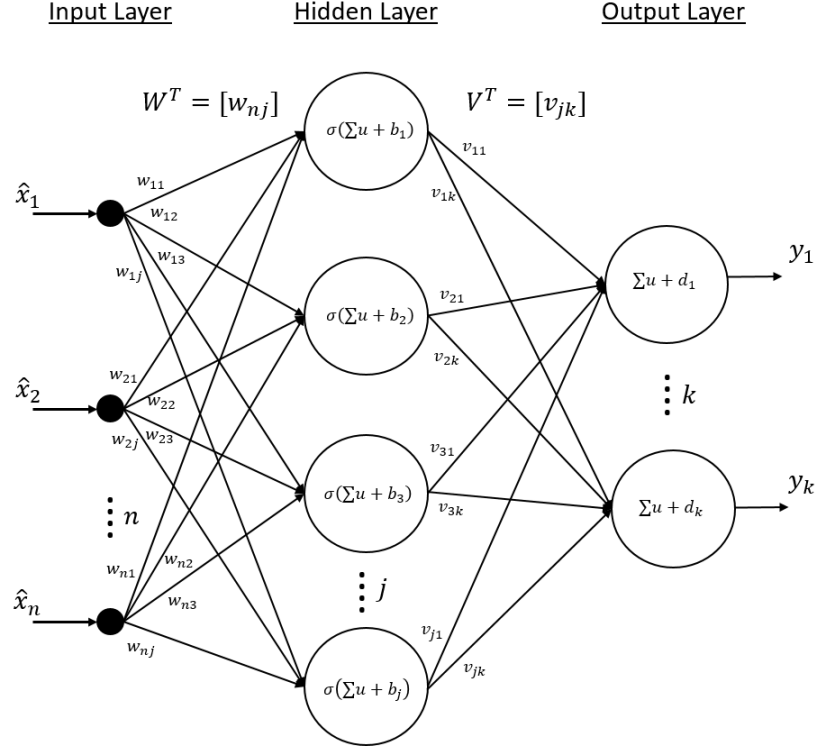


Figure 4.7. General schematic for the SHLS-NN.

4.3.2. Adaptive ANN augmentation architecture. The objective of the adaptive ANN augmentation is to aid the baseline NLDI control law in estimating inversion errors resulting from modeling uncertainties and wind disturbance. The adaptive ANN augmentation can be applied effectively by choosing an error signal which gives information about the linearization error of the baseline NLDI to drive the updating of the weights and biases inside the network. In this architecture, the adaptive ANN will attempt to reconstruct the linearization error, which can be then be used as an augmented value for the baseline NLDI control law.

The SHLS-NN is designed to augment the fast loop in the inner stability loop inversion of the baseline NLDI controller for which the architecture can be seen in Figure 4.8. Three SHLS-NNs are applied; one for each channel. In this architecture the adaptive

ANN estimates the inversion error in the fast rotational dynamics of the quadrotor UAV and augments the inversion with the estimation in an effort to achieve a fully linearized closed-loop response.

The SHLS-NN is designed to take five inputs, seven neurons in the hidden layer and one generated output per channel; the estimation of the inversion error in the fast loop. Inputs to the network are chosen to supply the network with sufficient information to reconstruct the inversion error (McFarland & Calise, 2000) (Kaneshige, Bull, & Totah, 2000).

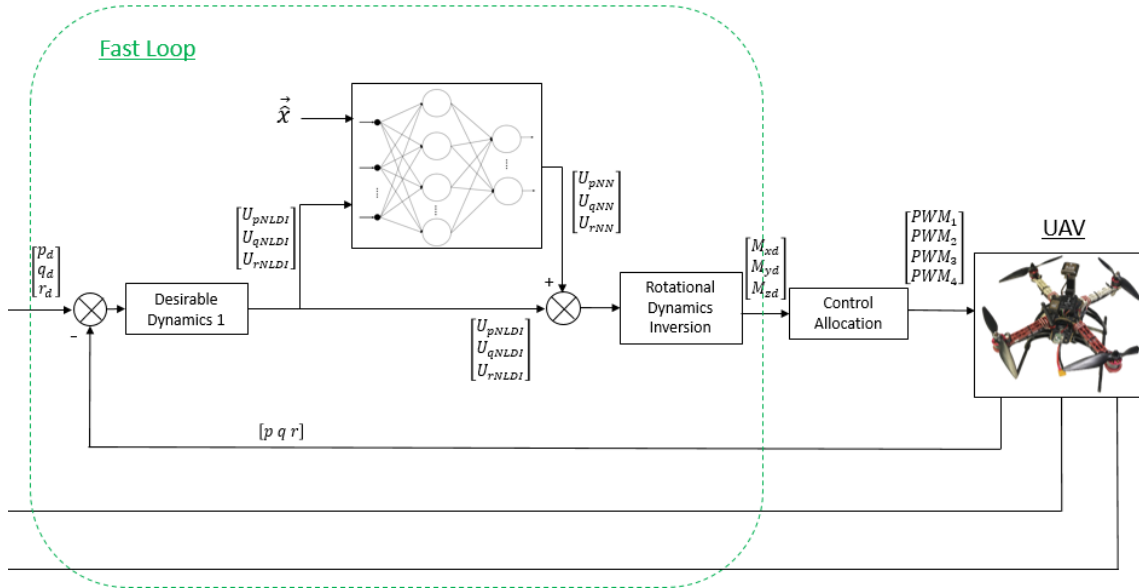


Figure 4.8. Fast loop augmentation by SHLS-NN adaptive ANN.

p-channel inputs:

$$\hat{\tilde{x}}_p = \left[\hat{p} \ \hat{q} \ \hat{r} \ \hat{p}_d \ \frac{1 - e^{-(U_{pNLDI} - U_{pNN'})}}{1 + e^{-(U_{pNLDI} - U_{pNN'})}} \right]^T \quad (4.55)$$

q-channel inputs:

$$\vec{\hat{x}}_q = \left[\hat{p} \ \hat{q} \ \hat{r} \ \hat{q}_d \ \frac{1 - e^{-(U_{qNLDI} - U_{qNN'})}}{1 + e^{-(U_{qNLDI} - U_{qNN'})}} \right]^T \quad (4.56)$$

r-channel inputs:

$$\vec{\hat{x}}_r = \left[\hat{p} \ \hat{q} \ \hat{r} \ \hat{r}_d \ \frac{1 - e^{-(U_{rNLDI} - U_{rNN'})}}{1 + e^{-(U_{rNLDI} - U_{rNN'})}} \right]^T \quad (4.57)$$

where \hat{p} , \hat{q} and \hat{r} are normalized sensor readings of the current roll, pitch and yaw rates respectively. \hat{p}_d , \hat{q}_d and \hat{r}_d are normalized commanded rates coming from the output of the slow loop. The last parameter in each channel feeds back information of the previous iteration output of the ANN which is fed through a sigmoidal function to bound the signal.

The error signal driving the updating of the weights and biases in the network is the output of the virtual controllers in the fast loop; U_{pNLDI} , U_{qNLDI} and U_{rNLDI} which are defined in Equation (4.2). Equation (4.2) shows that the output of the virtual controllers in the fast loop represent an error signal between desired and actual angular rates. This error is mainly generated by the uncertainties and disturbances in the fast loop.

When U_{pNLDI} , U_{qNLDI} and U_{rNLDI} are chosen as the signals that drive the adaptive update of the SHLS-NN, desired activation of the ANN is achieved. If U_{pNLDI} , U_{qNLDI} and U_{rNLDI} are equal to zero, which is the case for a perfect feedback linearized system, and the virtual controllers perfectly assure tracking of desired angular rates resulting in a zero error signal, the adaptation of the adaptive ANN is stopped and the current output of the SHLS-NN is maintained. The network is basically told that the current augmented value

is producing the desired result. When U_{pNLDI} , U_{qNLDI} and U_{rNLDI} are non-zero, the SHLS-NN will attempt to estimate and produce a control augmentation that will aid in reducing the error signal.

The learning rates are the main tuning parameters in the adaptive ANN architecture. For the simulation results shown in Chapters 5, the learning rates Γ_1 and Γ_2 defined in Equation (4.54) for the updating of the parameters of the SHLS-NN will be constant and equal to one. Since the learning rates are tuned specifically for each test flight in the implementation section, they will be defined for the individual test flight results shown in Chapters 7. The e-modification parameters required to promote boundedness of the network weights and bias terms (Equation (4.54)) are constant and equal to 0.02 for both the simulation (Chapter 5) and implementation (Chapter 7) environment.

The outputs of SHLS-NN; U_{pNN} , U_{qNN} and U_{rNN} defined in Equation (4.53), are the estimates of the inversion error in the fast loop and are added into the control loop in an attempt to cancel the inversion error. Assuming the virtual controllers consist of simple proportional controllers, the inputs to the inversion of the conservation of angular momentum equations presented in Equation (4.1) for a baseline NLDI control architecture with adaptive ANN augmentation can be expressed as:

$$\begin{bmatrix} U_p \\ U_q \\ U_r \end{bmatrix} = \begin{bmatrix} k_p(p_d - p) \\ k_q(q_d - q) \\ k_r(r_d - r) \end{bmatrix} + \begin{bmatrix} U_{pNN} \\ U_{qNN} \\ U_{rNN} \end{bmatrix} \quad (4.58)$$

4.3.3. Stability analysis. An important concern in the application of adaptive ANN to augment the baseline NLDI control laws is the boundedness of the outputs of the

SHLS-NN due to its self-adaptive configuration (Campa, Fravolini, Mammarella, & Napolitano, 2011). Stability bounds and error convergence of the NLDI control law with SHLSS-NN must be shown to assure stability for the fast loop.

The stability proof for the SHLS-NN applied to augment the NLDI control law is provided in (Calise, Lee, & Sharma, 2001) and can be applied to the fast loop augmentation presented in Section 4.3.2 by rewriting Equation (4.1) as:

$$\bar{M}_d = f^{-1}(\bar{\Omega}_b, \bar{U}_\Omega) \quad (4.59)$$

where $\bar{U}_\Omega = [U_p \ U_q \ U_r]^T$.

The inversion error in the fast loop can be presented as:

$$\bar{\Delta} = f^{-1}(\bar{\Omega}_b, \bar{U}_\Omega) - \hat{f}^{-1}(\bar{\Omega}_b, \bar{U}_\Omega) \quad (4.60)$$

where $\hat{f}^{-1}(\bar{\Omega}_b, \bar{U}_\Omega)$ is the modeling of the system dynamics, which in the case of an imperfect approximation will result in uncanceled nonlinear terms in the closed-loop of the fast loop.

Given Equations (4.3) and (4.58), the closed-loop dynamics in the fast loop can be expressed as:

$$\dot{\bar{\Omega}}_b = \bar{U}_\Omega + \bar{U}_{NN} + \bar{\Delta} + \dot{\bar{\Omega}}_{bd} \quad (4.61)$$

where $\bar{U}_{NN} = [U_{pNN} \ U_{qNN} \ U_{rNN}]^T$ are the augmented control values generated by the SHLS-NN and $\dot{\bar{\Omega}}_{bd} = [\dot{p}_d \ \dot{q}_d \ \dot{r}_d]^T$ are the desired closed-loop dynamics for the fast loop which are introduced by the stability proof (Calise, Sharma, & Corban, 2000).

The error dynamics of the fast loop can be expressed as:

$$\dot{\vec{e}}_{\Omega} = K_{\Omega} \vec{e}_{\Omega} - \vec{U}_{NN} - \vec{\Delta} \quad (4.62)$$

where $\vec{e}_{\Omega} = \vec{\Omega}_{bd} - \vec{\Omega}_b$ is the error in the fast loop and K_{Ω} is defined as:

$$K_{\Omega} = \begin{bmatrix} -K_p & 0 & 0 \\ 0 & -K_q & 0 \\ 0 & 0 & -K_r \end{bmatrix} \quad (4.63)$$

Using Equation (4.53) to express \vec{U}_{NN} , Equation (4.62) can be written as:

$$\dot{\vec{e}}_{\Omega} = K_{\Omega} \vec{e}_{\Omega} - (V^T \sigma(W^T \hat{x} + \hat{b}) + \hat{d}) - \vec{\Delta} \quad (4.64)$$

To assure the stability of the fast loop and effective tracking of desired angular rates, it must be shown that $\dot{\vec{e}}_{\Omega}$ in Equation (4.64) has stable dynamics and that the updating of the weights and biases in the SHLS-NN Equation (4.54) is bounded. In other words, it must be shown that all elements in \hat{W} and \hat{V} are bounded by an upper bound; \tilde{W} and \tilde{V} . If both the error dynamics are shown to be stable and the updating of the weights and biases has an upper bound, the adaptive ANN augmented NLDI will behave in a stable manner and can be considered stable.

(Calise, Lee, & Sharma, 2001) shows the proof of boundedness and stability for the application of SHLS-NN to augment a NLDI control law for inversion error estimation and correction. This is performed by constructing a candidate Lyapunov function that is shown to be negative semidefinite through the application of Lyapunov's direct method, which concludes the proof of stable error dynamics $\dot{\vec{e}}_{\Omega}$ and bounded weight matrices \hat{W} and \hat{V} . The negative semidefinite result holds for the application of the SHLS-NN to augment the fast loop as long the desired angular rates are bounded and

continuously differentiable, stable virtual controllers are designed in Equation (4.2), the uncertainties are matched; they are assumed to be in the bandwidth of the inputs to the system (Campa, Fravolini, Mammarella, & Napolitano, 2011), inputs to the SHLS-NN are bounded, the upper bounds on \hat{W} and \hat{V} are known, and the update law defined in Equation (4.54) is used to update the parameters in the SHLS-NN (Sharma & Calise, 2005).

4.3.4. Discrete adaptive ANN architecture. The SHLS-NN control architecture introduced in Sections 4.3.1 and 4.3.2 is defined in continuous time. However, for implementing this controller in a real-time environment, the control architecture must be provided in a discrete fashion. This can be obtained relatively simply since the SHLS-NN in Equation (4.53) does not contain continuous-time transfer functions and so can be converted to a sampled structure as:

$$y_q(iT_s + 1) = V^T(iT_s)\sigma\left(W^T(iT_s)\hat{x}(iT_s) + \hat{b}(iT_s)\right) + \hat{d}(iT_s) \quad (4.65)$$

where $T_s \in \Re$ is the sample time of the discrete system. A fast sample time is desired such that the discrete system converges to a continuous system for fast estimation and adaptation of uncertainties and disturbances. This is however limited to the processing capabilities of the hardware.

The assumption will be made that parameters in the network will remain constant in between sampling times such that all parameters shown in Figure 4.7 remain constant in between sampling times. The parameter update law defined in Equation (4.54) can be discretized as:

$$\begin{aligned}
\widehat{W}(iT_s + 1) &= -\Gamma_1 [\tilde{\hat{x}}(iT_s) e(iT_s) \widehat{V}^T(iT_s) \hat{\sigma}' + \lambda_1 |e(iT_s)| \widehat{W}(iT_s)] \\
\widehat{V}(iT_s + 1) &= -\Gamma_2 [(\hat{\sigma}(iT_s) - \hat{\sigma}' \widehat{W}^T(iT_s) \tilde{\hat{x}}(iT_s)) e(iT_s) \\
&\quad + \lambda_2 |e(iT_s)| \widehat{V}(iT_s)]
\end{aligned} \tag{4.66}$$

4.4. \mathcal{L}_1 Output-Feedback Adaptive Control Augmentation

Section 4.3 discussed the application of intelligent adaptive control to estimate and correct for the inversion error in the inner stability loop inversion of the NLDI control law. A different approach to compensate for external wind disturbance effects in the NLDI baseline control law is to apply model reference based adaptive controller. While the adaptive ANN is effectively an estimator applied to estimate the inversion error, which can subsequently be used for augmentation in an attempt to cancel the inversion errors, model reference based adaptive control does not estimate the error directly but focuses on generating an augmented control signal that will drive the system to follow a desired reference model. Section 2.1.3 motivated that the application of model reference adaptive control (MRAC) will result in a tradeoff between the fast tracking of reference values and the stability of the system which will degrade the overall robustness of the NLDI augmented system. \mathcal{L}_1 state-feedback adaptive control effectively decouples the fast adaption with the robustness of the system but requires full state-feedback to be effectively applied. The necessary state estimation will introduce estimation errors in the control law, which is undesired. The \mathcal{L}_1 output-feedback adaptive control offers both fast adaptation and robustness without the need for state estimation. \mathcal{L}_1 output-feedback adaptive control will therefore be applied to augment the NLDI baseline control law to aid the robustness of the system under wind disturbance conditions.

4.4.1. \mathcal{L}_1 output-feedback adaptive control architecture. The goal of the \mathcal{L}_1 augmentation is to assure the outputs of the closed-loop rotational dynamics of the baseline NLDI augmented system $y(t)$, will track the output $\hat{y}(t)$ of a desired minimum phase reference model $M(s)$, which consists of the baseline NLDI inner controller with a desired UAV quadrotor rotational dynamics model, to a provided bounded piecewise-continuous desired reference signal $r(t)$ in both the transient and steady state with all signals bounded (Geiser, Xargay, & Hovakimyan, 2011). The controller consists of an output predictor predicting the output of the quadrotor UAV system, an adaptation law which provides an estimate of uncertainties in the system by comparing the predicted outputs and the measured outputs, and finally a control law which generates a control input which will correct for the uncertainties in the system.

The quadrotor UAV rotational inner stability dynamics can be represented as a SISO transfer function (Hovakimyan & Cao, 2010):

$$y(s) = A(s)(u(s) + d(s)), \quad y(0) = 0 \quad (4.67)$$

where $y(s) \in \mathfrak{R}$ is the output of the system, $u(s) \in \mathfrak{R}$ is the input to the system, $A(s)$ is a strictly proper unknown transfer function and $d(s) \in \mathfrak{R}$ is the Laplace transform of unknown disturbances and uncertainties. The inverse Laplace transform of $d(s)$ results in $d(t) = f(t, y(t))$ where f is an unknown mapping subject to the Lipschitz continuity, which enforces a uniform continuity on the development of the uncertainties:

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|, \quad |f(t, y)| \leq L|y| + L_0 \quad (4.68)$$

where the existence of constants $L > 0$ and $L_0 > 0$ is guaranteed such that Equation

(4.68) holds $\forall t > 0$ and L, L_0 arbitrarily large.

Equation (4.67) can be written in state-space form as:

$$\begin{aligned}\dot{x}(t) &= A_m x(t) + b_m(u(t) + \sigma(t)), \quad x(0) = 0; \\ y(t) &= c_m^T x(t)\end{aligned}\tag{4.69}$$

where (A_m, b_m, c_m) is a minimal realization of the desired reference model $M(s)$. $M(s)$ is chosen controllable, observable and with A_m is Hurwitz. $\sigma(t)$ represents the time domain representation of the matched uncertainties, which are assumed to enter the dynamics with the same input matrix as the input command (Campa, Fravolini, Mammarella, & Napolitano, 2011), and can be modeled as the difference between the true inner stability dynamics $A(s)$ and the desired reference model $M(s)$, in the presence of additional uncertainties and disturbances $d(s)$:

$$\sigma(s) = \frac{(A(s) - M(s))u(s) + A(s)d(s)}{M(s)}\tag{4.70}$$

The unknown matched uncertainties $\sigma(t)$ is what differentiates the quadrotor UAV system response from the desired reference system $M(s)$ response and must therefore be corrected for with the application of \mathcal{L}_1 output-feedback adaptive control augmentation. The \mathcal{L}_1 output-feedback adaptive control applied to augment the baseline NLDI control law in the fast loop can be seen in Figure 4.9 for which the components are discussed next as derived in (Hovakimyan & Cao, 2010).

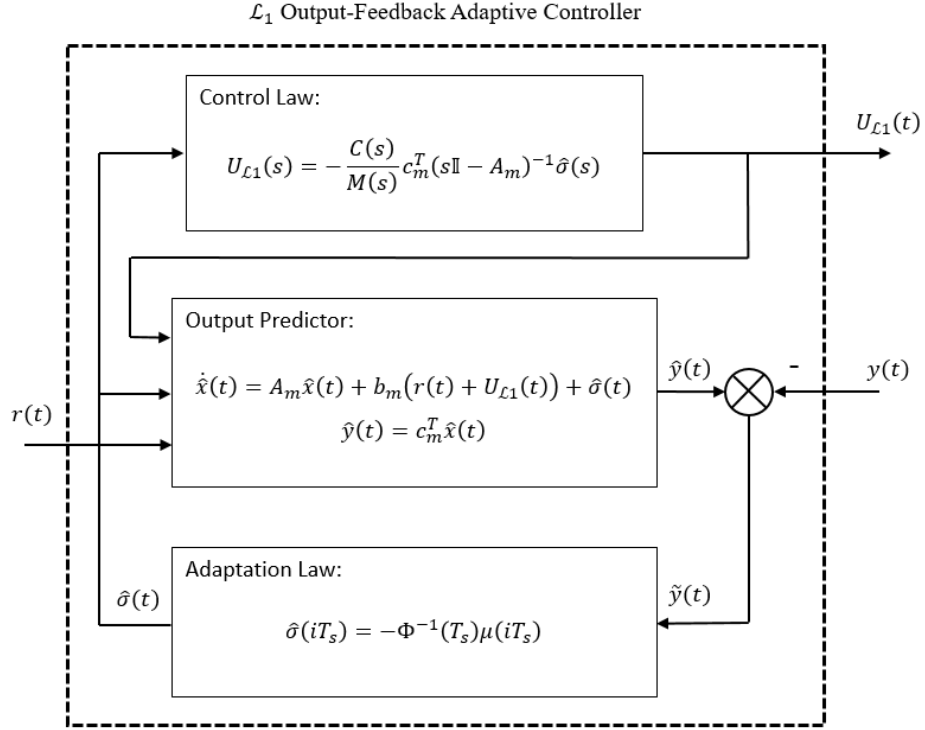


Figure 4.9. Augmenting \mathcal{L}_1 output-feedback adaptive controller.

Output predictor. The output predictor is used to provide a predicted output of the system based on the approximated uncertainties and the desired reference model such that it can be compared with the measured outputs. The stable and tracking output predictor can be expressed as:

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m \hat{x}(t) + b_m (r(t) + U_{\mathcal{L}1}(t)) + \hat{\sigma}(t), \quad \hat{x}(0) = 0; \\ \hat{y}(t) &= c_m^T \hat{x}(t) \end{aligned} \tag{4.71}$$

where $r(t) \in \mathfrak{R}$ is a reference input to the output predictor, $U_{\mathcal{L}1}(t) \in \mathfrak{R}$ is the control signal generated by the control law of the \mathcal{L}_1 output-feedback adaptive controller and $\hat{\sigma}(t) \in \mathfrak{R}$ is the approximation of the uncertainties in Equation (4.70) supplied by the adaptation law. Notice that while the uncertainties in the quadrotor UAV system are

assumed to be matched in Equation (4.69), the uncertainties in the output predictor Equation (4.71) are unmatched, which is a key feature and is shown to effectively cancel the uncertainties in the plant (Cao & Hovakimyan, 2008).

Adaptation law. The adaptation law updates the approximation of the uncertainties in Equation (4.70) according to the following discrete rule with discrete sample time $T_s \in \mathfrak{R}$:

$$\hat{\sigma}(iT_s) = -\Phi^{-1}(T_s)\mu(iT_s) \quad (4.72)$$

with,

$$\Phi(T_s) \triangleq \int_0^{T_s} e^{\Lambda A_m \Lambda^{-1}(T_s - \tau)} \Lambda d\tau \quad (4.73)$$

$$\Lambda \triangleq \begin{bmatrix} c_m^T \\ D\sqrt{P} \end{bmatrix}$$

where $P = P^T > 0$, and $P \in \mathfrak{R}^{n \times n}$ is the positive definite solution to the algebraic Lyapunov equation:

$$A_m^T P + P A_m = -Q, \text{ for arbitrary } Q = Q^T > 0 \quad (4.74)$$

which exists since A_m is Hurwitz.

$D \in \mathfrak{R}^{(n-1) \times n}$ in Equation (4.73) is a matrix that contains the null space of $c_m^T(\sqrt{P})^{-1}$, that is:

$$D \left(c_m^T(\sqrt{P})^{-1} \right)^T = 0 \quad (4.75)$$

$\mu(iT_s)$ in Equation (4.72) is defined as:

$$\mu(iT_s) = e^{\Lambda A_m \Lambda^{-1} T_s} \mathbf{1}_1 (\hat{y}(t) - y(t)) \quad (4.76)$$

where $\mathbf{1}_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^n$.

As can be seen, update law Equation (4.72) is defined discrete while the rest of the controller and quadrotor UAV model are modeled in continuous time. The assumption is therefore made that $\hat{\sigma}(iT_s)$ remains constant in between sample times:

$$\hat{\sigma}(t) = \hat{\sigma}(iT_s), \quad t \in [iT_s, (i+1)T_s] \quad (4.77)$$

The rate of adaptation is limited to the chosen sample time T_s . For the desired fast adaptation, the sampling time must be decreased, which will reduce the prediction error (Geiser, Xargay, & Hovakimyan, 2011). The sampling time however is often restricted by hardware limitations.

Control law. Now that the approximation of uncertainty in the system is provided by the adaptation law, a control law must be constructed to correct for this error through augmentation of the quadrotor UAV baseline inner loop NLDI control law. The \mathcal{L}_1 output-feedback control law consists of the estimation of the uncertainty in Equation (4.72) passed through a low pass filter (Cao & Hovakimyan, 2009). The augmentation value provided by the control law of the \mathcal{L}_1 output-feedback adaptive controller can be expressed as:

$$U_{\mathcal{L}_1}(s) = -\frac{C(s)}{M(s)} c_m^T (s\mathbb{I} - A_m)^{-1} \hat{\sigma}(s) \quad (4.78)$$

where $M(s)$ is the transfer function of the desired reference model and $C(s)$ is a strictly

proper filter with $C(0) = 1$ and relative degree d_r , which is the difference between the degree of the denominator and numerator of the filter. d_r is defined as:

$$1 < d_r \leq n_r \quad (4.79)$$

where n_r is the relative degree of $M(s)$ in Equation (4.69).

The filtering of the estimation of the uncertainty $\hat{\sigma}(s)$ in the system effectively decouples the fast adaptation from the robustness and assures that the control input sent to the system will remain inside a desired bandwidth. (Geiser, Xargay, & Hovakimyan, 2011). The upper bound of the bandwidth is usually associated with the available bandwidth of the actuators. The control input in Equation (4.78) will therefore result in a cancellation of the uncertainties inside the bandwidth of filter $C(s)$ and actuators. It is desirable to apply the highest bandwidth possible to achieve optimal results for the estimation of uncertainties for cancellation using the control law.

For the case that the measured outputs from the quadrotor UAV system track the outputs of the state predictor, that is $\hat{y}(t) = y(t)$, it can be seen that the adaptation law in Equation (4.72) will output zero signifying that no uncertainties are present in comparison between the true system and reference model. This results in the augmented control input from the \mathcal{L}_1 output-feedback adaptive controller in Equation (4.78) to be zero, which indicates that the \mathcal{L}_1 output-feedback adaptive controller does not augment the baseline NLDI control law for a system tracking the desired reference model perfectly. Only when there is uncertainty in the system will the \mathcal{L}_1 output-feedback adaptive controller attempt to make the system more robust.

4.4.2. Stability analysis. In order to verify the stability of the \mathcal{L}_1 output-

feedback adaptive controller, boundedness of all values together with the convergence of the output of the reference model and actual system must be shown. To ensure stability of the closed-loop performance of the \mathcal{L}_1 output-feedback adaptive controller, strictly proper filter $C(s)$ and desired reference model $M(s)$ must be designed such that (Cao & Hovakimyan, 2008):

$$H(s) = \frac{A(s)M(s)}{C(s)A(s) + (1 - C(s))M(s)} \quad (4.80)$$

is stable and

$$\|H(s)(1 - C(s))\|_{\mathcal{L}_1} < 1 \quad (4.81)$$

where $\| \cdot \|_{\mathcal{L}_1}$ is the \mathcal{L}_1 norm defined as (Hovakimyan & Cao, 2010):

$$\|f\|_{\mathcal{L}_1} \triangleq \int_0^\infty \|f(\tau)\| d\tau < \infty \quad (4.82)$$

Now that the boundedness of the system is assured through the correct design of $C(s)$ and $M(s)$, it must be shown that the \mathcal{L}_1 output-feedback adaptive controller converges and the uncertainty is adapted correctly. This can be shown by considering the ideal closed-loop model performance:

$$y_{ideal}(s) = M(s)(u_{ideal}(s) + r(s) + \sigma_{ideal}(s)) \quad (4.83)$$

$$\sigma_{ideal}(s) = \frac{(A(s) - M(s))u_{ideal}(s) + A(s)d_{ideal}(s)}{M(s)} \quad (4.84)$$

$$u_{ideal}(s) = -C(s)\sigma_{ideal}(s) \quad (4.85)$$

Note that for the ideal closed-loop model, the uncertainties $\sigma_{ideal}(s)$ can only be corrected for inside the bandwidth of the filter $C(s)$, as can be seen when Equation (4.85) is inserted in Equation (4.83). It must now be shown that the designed \mathcal{L}_1 output-feedback adaptive controller converges to the ideal closed-loop reference model. (Cao & Hovakimyan, 2008) shows the proof for this convergence:

$$\lim_{T \rightarrow 0} (\|\hat{y}(t) - y(t)\|_{\mathcal{L}_1}) = 0 \quad (4.86)$$

which assures the convergence of the output predictor Equation (4.71) and the system output indicating the correction for uncertainties in the system,

$$\lim_{T \rightarrow 0} (\|y(t) - y_{ideal}(t)\|_{\mathcal{L}_1}) = 0 \quad (4.87)$$

which assures the convergence of the system output and the ideal system in Equation (4.83), confirming the obtained desired dynamic characteristics, and finally

$$\lim_{T \rightarrow 0} (\|u(t) - u_{ideal}(t)\|_{\mathcal{L}_1}) = 0 \quad (4.88)$$

which confirms the convergence of the control law in Equation (4.78) to the ideal control law presented in Equation (4.85), which will effectively correct for the uncertainties in the system.

Although the proof shows the steady-state desired convergence of Equations (4.87) and (4.88), the transient tracking of Equations (4.87) and (4.88) is uniformly

bounded by a constant proportional to the sample time T_s . In other words, the transient tracking response can be improved by reducing the sample time in the controller.

4.4.3. \mathcal{L}_1 output-feedback adaptive control augmentation

architecture. \mathcal{L}_1 output-feedback adaptive control augmentation can be applied to correct for the uncertainties in the inner stability loop inversion caused by wind disturbance. This can be achieved by evaluating the tracking performance of the attitude angles and by generating an augmented control signal to be added in the fast loop of the inner stability controller, assuring the fast correction of uncertainties. The \mathcal{L}_1 output-feedback adaptive controller can be applied to aid in the tracking performance of the attitude angles. The \mathcal{L}_1 output-feedback adaptive control augmentation evaluates the tracking performance of the slow loop of the NLDI inner stability loop compared to a reference model and creates an augmented value to be added to the fast loop similar as in (Leman, Xargay, Dullerud, & Hovakimyan, 2009). This augmented value will attempt to correct for the fast dynamic uncertainties in the inner loop by augmenting the fast loop. The fast loop augmentation by the \mathcal{L}_1 output-feedback adaptive controller can be seen in Figure 4.10.

Recall that the output predictor resembles the UAV quadrotor dynamics with the application of the inner NLDI loop. In the same fashion as for the inner NLDI loop, the reference signals provided to the \mathcal{L}_1 controller are therefore the desired attitude angles; $[\theta_d \ \phi_d \ \psi_d]^T$. The output of the output predictor will signify the predicted tracking response of the desired attitude angles by the quadrotor dynamics with inner NLDI loop control. The output of the output predictor is compared with the actual measured response of the quadrotor UAV system; $[\theta \ \phi \ \psi]^T$. The adaptation and control law in the \mathcal{L}_1

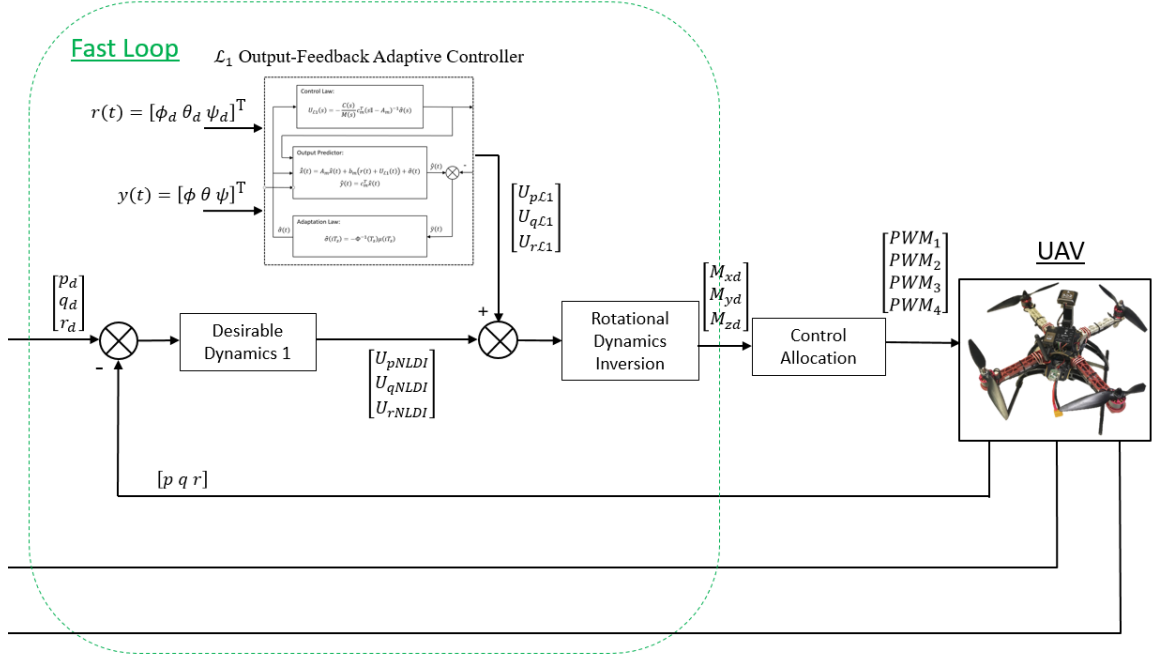


Figure 4.10. Fast loop augmentation by \mathcal{L}_1 output-feedback adaptive control.

output-feedback adaptive controller will generate an augmented control signal based on the difference between the output predictor and measured response; $[U_{pL1} \ U_{qL1} \ U_{rL1}]^T$.

Three \mathcal{L}_1 output-feedback adaptive controllers are used to augment the inner stability loop, one for each channel. Assuming the virtual controllers consist of a simple proportional controller, the inputs to the inversion of the conservation of angular momentum equations presented in Equation (4.1) for a baseline NLDI control architecture with \mathcal{L}_1 output-feedback adaptive controller can be expressed as:

$$\begin{bmatrix} U_p \\ U_q \\ U_r \end{bmatrix} = \begin{bmatrix} k_p(p_d - p) \\ k_q(q_d - q) \\ k_r(r_d - r) \end{bmatrix} + \begin{bmatrix} U_{pL1} \\ U_{qL1} \\ U_{rL1} \end{bmatrix} \quad (4.89)$$

Reference model design. The reference model signifying the quadrotor UAV rotational dynamics with inner NLDI loop can be modeled as a second order minimum

phase stable transfer function:

$$M(s) = \frac{\omega_{n_m}^2}{s^2 + 2\zeta_m \omega_{n_m} s + \omega_{n_m}^2} \quad (4.90)$$

where $\omega_{n_m} \in \mathfrak{R}$ is the natural frequency and $\zeta_m \in \mathfrak{R}$ the damping ratio of the desired reference model. This reference system will track the desired inputs as shown by the final value theorem under arbitrary step response:

$$\lim_{s \rightarrow 0} s \frac{\omega_{n_m}^2}{s^2 + 2\zeta_m \omega_{n_m} s + \omega_{n_m}^2} \cdot \frac{\delta}{s} = \delta \quad (4.91)$$

where $\delta \in \mathfrak{R}$ is an arbitrary step value. Since the reference model converges to the desired reference values, the natural frequency ω_{n_m} and damping ratio ζ_m can be selected for the desired transient characteristics of the reference model.

For the simulation results presented in Chapter 5, the reference model is designed with $\omega_{n_m} = 12$ rad/s and $\zeta_m = 1.3$. This design will result in an overdamped, relatively fast response to reflect the desired response of the UAV quadrotor system with NLDI inner loop in the simulation environment. The rise time and settling time of the reference model are 0.4 sec and 0.7 sec respectively.

For the implementation of the reference model in the real-time system, for which results are presented in Chapter 7, the reference model is designed to mimic the nominal step response of the quadrotor UAV system in flight. The reference model is designed with $\omega_{n_m} = 25$ rad/s and $\zeta_m = 1.3$ to mimic the nominal step response of the baseline NLDI fast loop, which will give a rise time of 0.195 sec and a settling time of 0.35 sec.

The reference model transfer function in Equation (4.90) can be converted into

state-space format for application in Equation (4.71) as:

$$A_m = \begin{bmatrix} 0 & 1 \\ -\omega_{n_m}^2 & -2\zeta_m\omega_{n_m} \end{bmatrix}$$

$$b_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.92)$$

$$c_m = [\omega_{n_m}^2 \quad 0]$$

Low-pass filter design. The low-pass filter applied to filter the estimation of the uncertainties in the inner stability loop can be modeled using a second order transfer function which satisfies Equation (4.77):

$$C(s) = \frac{\omega_{n_c}^2}{s^2 + 2\zeta_c\omega_{n_c}s + \omega_{n_c}^2} \quad (4.93)$$

where $\omega_{n_c} \in \Re$ is the natural frequency and $\zeta_c \in \Re$ the damping ratio of the low-pass filter.

When designing the filter, attention must be placed on selecting the bandwidth. As discussed in Section 4.4.1, the upper bound of the bandwidth is often linked to the upper bound of the actuators. Besides the bandwidth, the design of $C(s)$ must also assure that the resonant frequency of the filter is ideally outside of the bandwidth, and if it is located inside the bandwidth, the resonance peak must be negligibly small to ensure that the filter solely filters the incoming signal and does not cause amplification.

For the simulation results shown in Chapter 5, the filter is designed with $\omega_{n_c} = 300$ rad/s and $\zeta_m = 0.7$. The filter specifics can be visualized with the Bode plot in Figure 4.11. The Bode plot shows a filter bandwidth of 268 rad/s with no resonant peak

indicating that the filter does not amplify the incoming signal. The application of this designed filter will allow for the correction of uncertainties with a natural frequency below 268 rad/s.

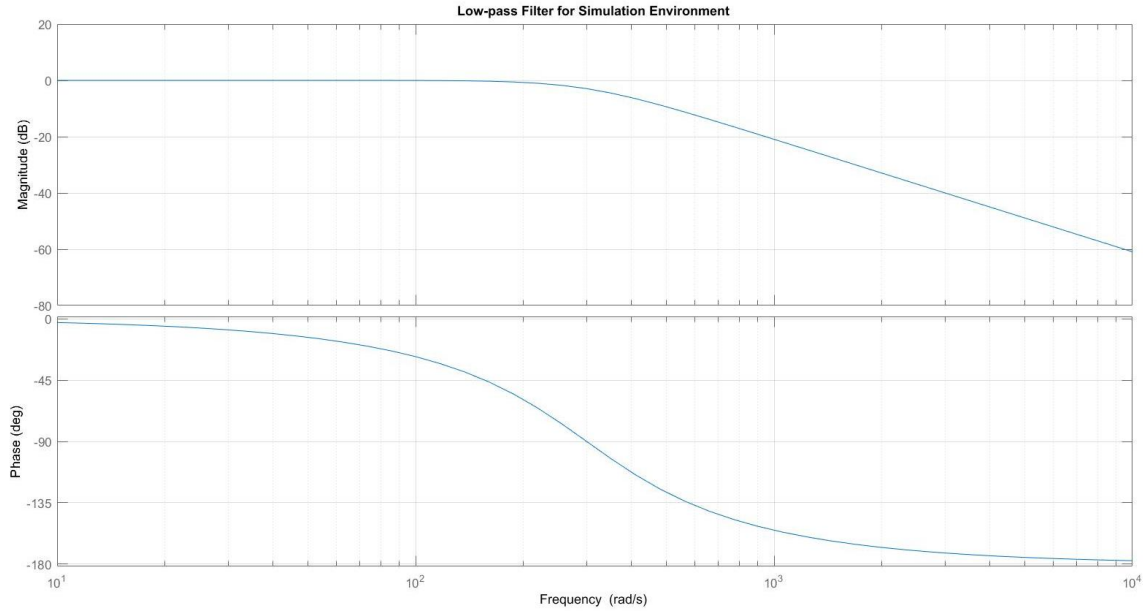


Figure 4.11. Bode plot for the low-pass filter designed for the simulation environment.

For the implementation of the low-pass filter in the UAV quadrotor system, for which results are shown in Chapter 7, the filter is designed as with $\omega_{n_c} = 120$ rad/s and $\zeta_m = 0.7$. The filter specifics can be visualized with the Bode plot in Figure 4.12. The Bode plot shows a filter bandwidth of 105 rad/s with no resonant peak indicating that the filter does not amplify the incoming signal. The application of this designed filter will allow for the correction of uncertainties with a natural frequency below 105 rad/s.

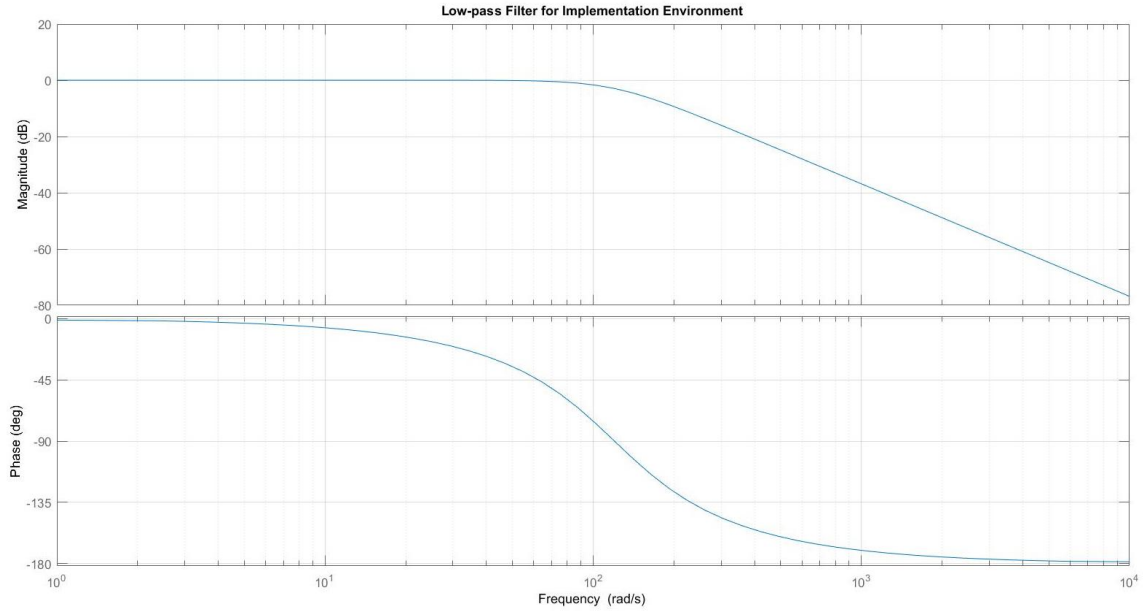


Figure 4.12. Bode plot for the low-pass filter designed for the implementation environment.

4.4.4. Discrete \mathcal{L}_1 output-feedback adaptive control architecture. As discussed in Section 2.3, it is important to consider the sample time for the implementation of the \mathcal{L}_1 output-feedback adaptive controller on the onboard computer of the quadrotor UAV system. Section 4.4.3 showed the synthesis of the \mathcal{L}_1 output-feedback adaptive controller includes continuous-time second order transfer functions for the reference model and low-pass filter which cannot be implemented in a discrete environment. These second order transfer functions must therefore be converted to a discrete definition using the sample time to allow for the application in a real-time environment.

A generic continuous-time, time-invariant state-space model can be discretized using fictitious samplers and holding devices assuming the input changes only at equally spaced sampling instances (Ogata, 1995):

$$x((k+1)T_s) = Gx(kT_s) + Hu(kT_s), \quad \hat{x}(0) = 0; \quad (4.94)$$

$$y(kT_s) = Cx(kT_s)$$

where matrix $G \in \mathfrak{R}^{n \times n}$ is the discretized continuous state matrix A , $H \in \mathfrak{R}^{n \times m}$ is the discretized continuous input matrix B and $T_s \in \mathfrak{R}$ is the sample time. Matrices G and H are defined as:

$$G = e^{AT_s}$$

$$H = \int_0^{T_s} e^{A(T_s-s)} B ds \quad (4.95)$$

The above shown discretization of a continuous-time state-space model can be used to discretize the various components in the \mathcal{L}_1 output-feedback adaptive controller as shown in (Wang, Kharisov, & Hovakimyan, 2014) and (Jafarnejadsani, Lee, & Hovakimyan, 2017).

Discretized output predictor. The continuous-time output predictor in Equation (4.71) can be discretized using the definitions in Equations (4.94) and (4.95) as:

$$\begin{aligned} \hat{x}((k+1)T_s) &= \hat{A}_m \hat{x}(kT_s) + \hat{B}_1 \left(r(kT_s) + \hat{U}_{\mathcal{L}_1}(kT_s) \right) \\ &+ \hat{B}_2 \hat{\sigma}(kT_s), \quad \hat{x}(0) = 0; \end{aligned} \quad (4.96)$$

$$\hat{y}(kT_s) = c_m^T \hat{x}(kT_s)$$

where $r(kT_s)$ is the discretized reference input to the output predictor, and \hat{A}_m , \hat{B}_1 and \hat{B}_2 are defined as:

$$\begin{aligned}\hat{A}_m &= e^{A_m T_s} \\ \hat{B}_1 &= \int_0^{T_s} e^{A_m(T_s-s)} b_m ds \\ \hat{B}_2 &= \int_0^{T_s} e^{A_m(T_s-s)} ds\end{aligned}\tag{4.97}$$

Note that no conversion is needed for output matrix c_m^T when discretized from continuous-time.

Discretized adaptation law. The adaptation law which updates the approximation of the uncertainties in the system presented in Equation (4.72) is already presented in a piecewise discrete rule and can therefore be used in the discretized \mathcal{L}_1 output-feedback adaptive controller.

Discretized control law. The continuous-time \mathcal{L}_1 output-feedback adaptive control law presented in Equation (4.78) can be discretized by first rewriting Equation (4.78) as:

$$U_{\mathcal{L}_1}(s) = -W(s)\hat{\sigma}(s)\tag{4.98}$$

where

$$W(s) = \frac{C(s)}{M(s)} c_m^T (s\mathbb{I} - A_m)^{-1}\tag{4.99}$$

In order to discretize the control law in Equation (4.98), the continuous-time transfer function in Equation (4.98) can first be converted to a continuous-time state-space representation:

$$\dot{x}_W(t) = A_W x_W(t) + B_W \hat{\sigma}(t), \quad x_W(0) = 0; \quad (4.100)$$

$$y_W(t) = U_{L1}(t) = -C_W x_W(t)$$

where the estimation of the uncertainty $\hat{\sigma}(t)$ is the input to system, the state of the system is described by x_W , and the output of the system is the control signal that will augment the fast loop of the NLDI baseline controller; $U_{L1}(t)$. The continuous-time state-space system can subsequently be converted to a discrete state-space representation using Equations (4.94) and (4.95):

$$x_W((k+1)T_s) = \hat{A}_W x_W(kT_s) + \hat{B}_W \hat{\sigma}(kT_s), \quad x_W(0) = 0; \quad (4.101)$$

$$y_W(kT_s) = U_{L1}(kT_s) = -C_W x_W(kT_s)$$

where $U_{L1}(kT_s)$ is the discretized control signal augmenting the fast loop of the baseline NLDI controller in the implementation environment, and \hat{A}_W and \hat{B}_W are defined as:

$$\begin{aligned} \hat{A}_W &= e^{A_W T_s} \\ \hat{B}_W &= \int_0^{T_s} e^{A_W(T_s-s)} B_W ds \end{aligned} \quad (4.102)$$

5. Simulation Analysis

Before implementing the developed control architectures in a quadrotor UAV testbed, the proposed control algorithms are analyzed in a simulation environment for an initial performance evaluation. This chapter introduces the simulation environment in which the developed mathematical models in Chapter 3 and control architectures in Chapter 4 are numerically solved to evaluate the performance of the discussed control

algorithms. A performance metric is presented to objectively evaluate the performance of the different control algorithms for wind disturbance conditions inside a considered wind envelope using Monte Carlo analysis.

5.1. Quadrotor UAV Simulation Environment

A high fidelity simulation environment was designed in MATLAB/Simulink to test the developed control algorithms in Chapter 4 using the mathematical model for the UAV quadrotor and the mathematical model for the wind environment developed in Chapter 3. A DJI Flamewheel F330 quadrotor research testbed was used to implement the developed adaptive control algorithms, and so the simulation environment was built around a simulation model of the Flamewheel quadrotor. The simulation environment, shown in Figure 5.1, consists of the mathematical Flamewheel model, a sensor and filtering model which simulates on-board sensors and estimation operations, a waypoint navigation model which produces autonomous waypoints for navigation, a control laws block in which the control laws are implemented and a wind model block which simulates external wind velocities and accelerations. Sections 5.1.1, 5.1.2, 5.1.3 and 5.1.4 will present the above mentioned blocks in more detail.

5.1.1. DJI flamewheel F330 model. Figure 5.2 shows the ‘Flamewheel’ block that contains a simulation model for the DJI Flamewheel F330. At its core, the Flamewheel is modeled around governing Equations (3.2), (3.3), (3.4) and (3.5). This includes the modeled wind forces and moments found in Sections 3.2.2 and 3.2.3. Wind accelerations are used to generate forces according to Equation (3.31). The wind velocity defined in the inertial earth reference frame is added to the quadrotor UAV velocity to include wind effects as seen in Equation (3.18).

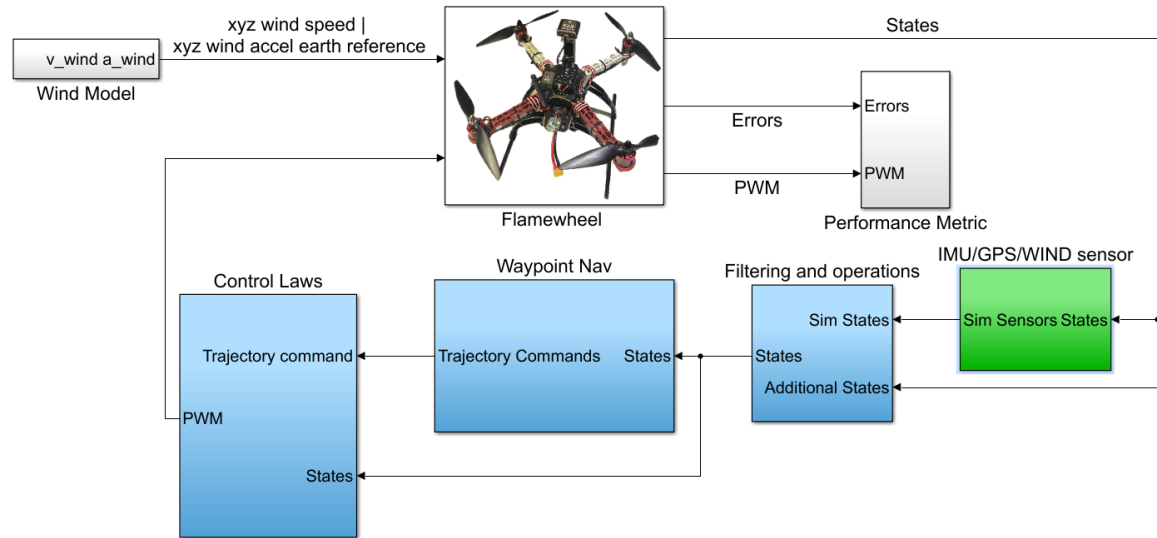


Figure 5.1. The DJI Flamewheel F330 simulation environment.

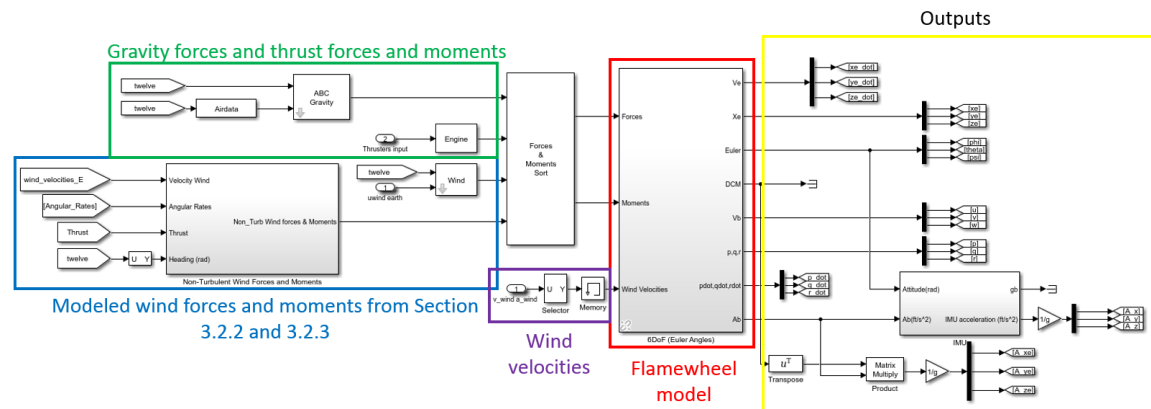


Figure 5.2. DJI Flamewheel F330 simulation model.

The various required parameters to completely define the simulation model for the DJI Flamewheel and wind induced forces and moments as outlined in Chapter 3 are presented in Table 5.1.

Table 5.1

Quadrotor UAV and Blade Flapping Model Simulation Parameters

Quadrotor UAV Model		Blade Flapping Model	
m	2.2lb	ω_p	62.83rad/s
L	6.5ft	R_{prop}	7.5in
h	1.1in	θ_0	7.5deg
$I_{xx} = I_{yy}$	0.0132slug·ft ²	θ_{tw}	2deg
I_{zz}	0.0230slug·ft ²	K_β	2.04 lb·in/rad
$I_{xz} = I_{xy} = I_{yz}$	0		
g	32.17ft/s ²		

5.1.2. Sensor and filter model. The ‘IMU/GPS/WIND’ block in Figure 5.1 simulates the noisy measurements of the global positioning system (GPS), inertial measurement unit (IMU) and a generic wind sensor that supplies wind velocities and accelerations as can be seen in Figure 5.3. The obtained simulation results shown in this chapter do not included added sensor noise and bias for the simulated GPS, IMU and wind sensor models in order to evaluate the control algorithms exclusively under wind disturbance conditions. The simulation frequency throughout the simulation environment, including the frequency of all simulated sensors, is defined at $T_s = 500Hz$.

The simulation of the post processing of IMU and GPS data is performed in the ‘Filtering and operations’ block found in Figure 5.1. GPS readings are converted to positions and velocities in the inertial earth reference frame and Euler angles are estimated using the kinematic equations. This block provides the simulation with the states of the Flamewheel during each time instant. The filtering and operations simulation is shown in Figure 5.4.

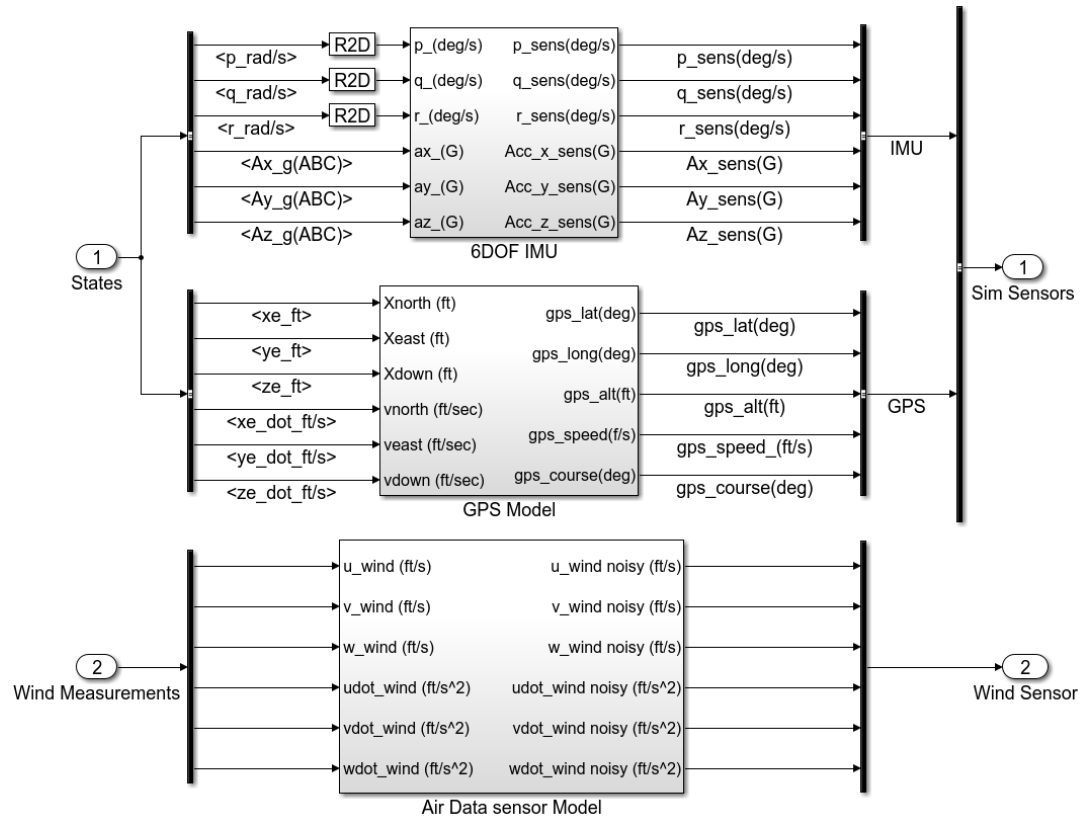


Figure 5.3. Sensor simulation model.

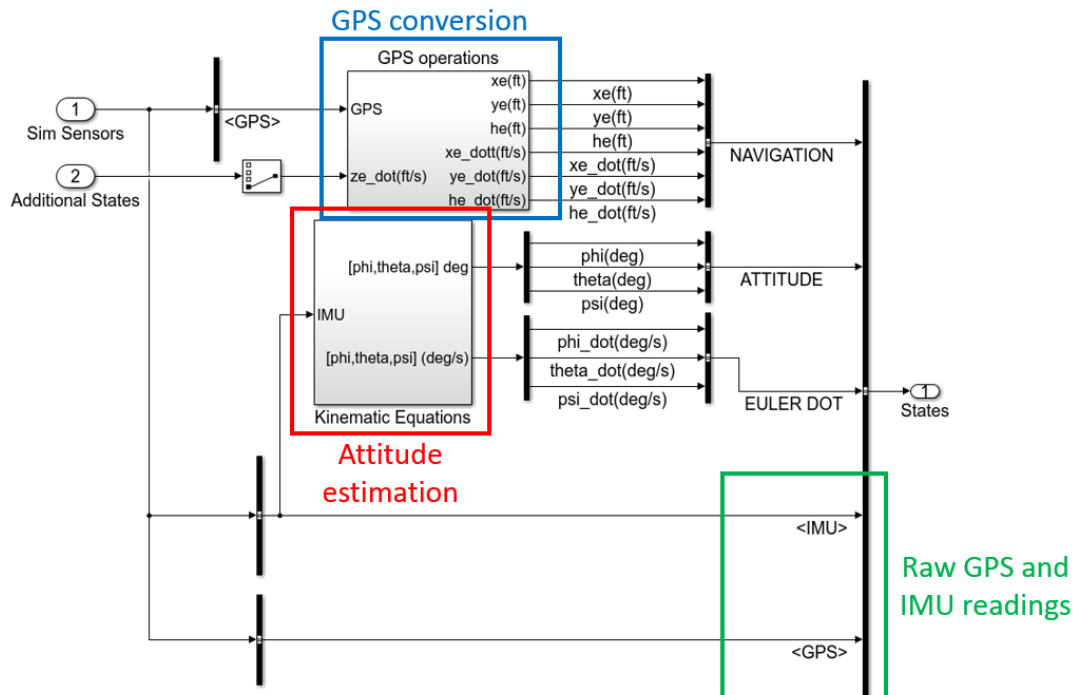


Figure 5.4. Filtering and operations simulation model.

5.1.3. Waypoint navigation and control architecture model. The ‘Waypoint Nav’ block in Figure 5.1 generates the desired waypoint commands for the DJI Flamewheel quadrotor simulation to track for a full autonomous simulation environment as can be seen in Figure 5.5.

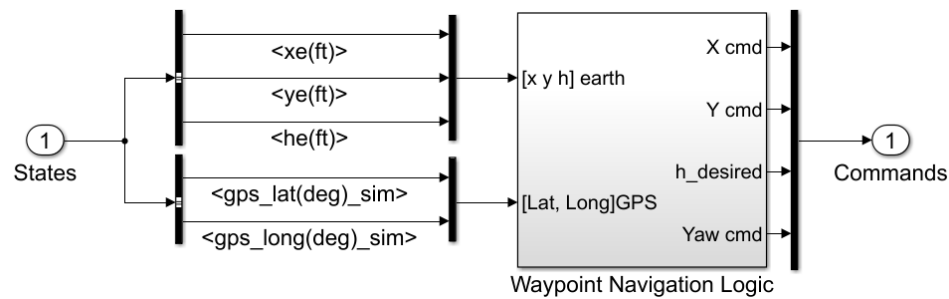


Figure 5.5. Waypoint navigation simulation model.

The ‘Control Laws’ block in Figure 5.1 models the baseline NLDI, extended NLDI, adaptive ANN and \mathcal{L}_1 adaptive controller. The control laws architecture can be seen in Figure 5.6 where the extended NLDI is modeled in the ‘Outer Tracking Loop’ and ‘Inner Stability Loop’. The SHLS-NN and \mathcal{L}_1 adaptive controllers are both modeled in the ‘Inner Stability Loop’. The baseline NLDI virtual controllers are tuned as shown in Table 5.2 to obtain the desired dynamic characteristics.

The simulation model for the SHLS-NN has been provided by previous research efforts performed at the West Virginia University (Campa, Fravolini, & Napolitano, 2002). The Simulink model containing the simulation model for the SHLS-NN is shown in Figure 5.7.

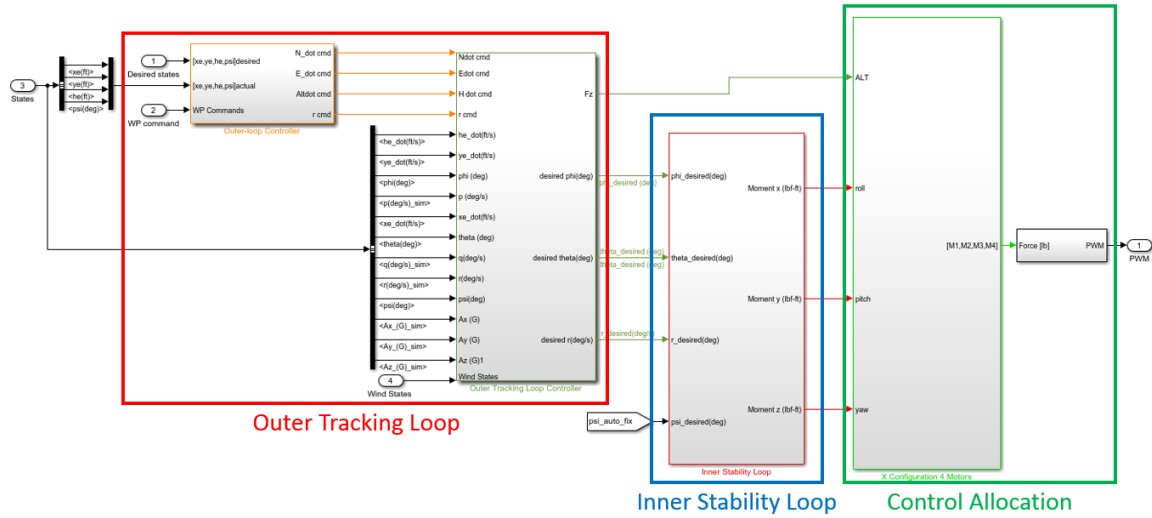
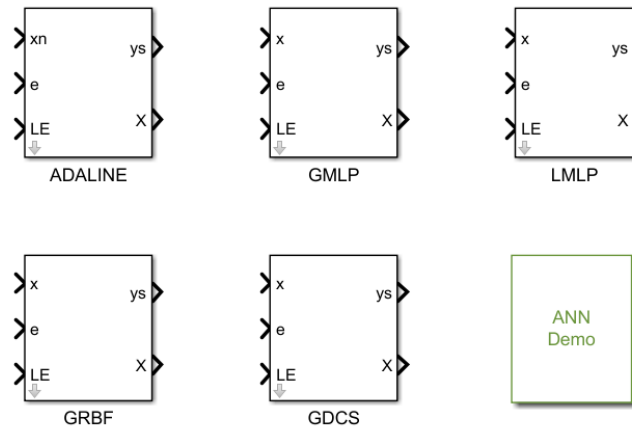


Figure 5.6. Control laws architecture simulation model.

Table 5.2

Baseline NLDI Simulation Virtual Controller Tuned Gains

Inner Stability Fast Loop		
Controller	PID gains	Equation (4.2)
U_p, U_q	[0.4 0.1 0]	
U_r	[0.3 0 0]	
Inner Stability Slow Loop		
Controller	PID gains	Equation (4.5)
U_θ, U_ϕ	[3.1 1 0]	
U_ψ	[1.4 0 0]	
Outer Tracking loop		
Controller	PID gains	Equation (4.19)
$K_{\dot{x}}, K_{\dot{y}}$	[3 0 0.4]	
$K_{\dot{z}}$	[17.5 2.5 0]	
K_x, K_y	[0.1 0 0]	
K_z	[1 0 0]	



Adaptive Neural Network Library, (Matlab R11.1 through R14)
 Giampiero Campa, West Virginia University, July 2007

Figure 5.7. Adaptive ANN simulation model.

5.1.4. Wind model. The ‘Wind Model’ block in Figure 5.1 contains a simulation model for the wind environment, as shown in Figure 5.8. Wind magnitude and direction are defined by the user inside the ‘Wind’ block in Figure 5.8. The magnitude and direction are used to define the wind velocity and acceleration vectors in the inertial earth reference frame.

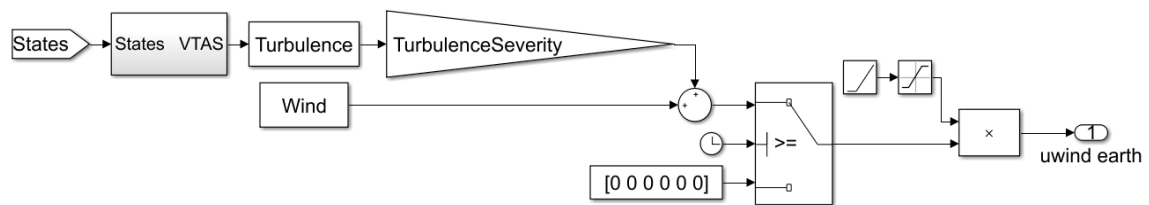


Figure 5.8. Simulated wind environment.

Simulated atmospheric turbulence is added to the wind velocity and accelerations vectors. The ‘Turbulence’ block contains filters that take in white noise to produce a colored noise turbulence representation in compliance with the PSD functions defined in

Equations (3.19), (3.20) and (3.21) (Rauw, 2001). These filters can be modeled in the simulation environment as:

$$\Delta W_X(\sigma) = K_{turbulence\ severity} \sqrt{\frac{2L_u}{V}} \frac{1}{1 + \frac{L_u}{V} j\sigma} \quad (5.1)$$

$$\Delta W_Y(\sigma) = K_{turbulence\ severity} \sqrt{\frac{L_v}{V}} \frac{1 + \sqrt{3} \frac{L_v}{V} j\sigma}{\left(1 + \frac{L_v}{V} j\sigma\right)^2} \quad (5.2)$$

$$\Delta W_Z(\sigma) = K_{turbulence\ severity} \sqrt{\frac{L_w}{V}} \frac{1 + \sqrt{3} \frac{L_w}{V} j\sigma}{\left(1 + \frac{L_w}{V} j\sigma\right)^2} \quad (5.3)$$

where $\Delta W_X(\sigma)$, $\Delta W_Y(\sigma)$ and $\Delta W_Z(\sigma) \in \Re$ are the simulated components of added turbulence originally defined in Equation (3.17) as a function of white noise input $\sigma \in \Re$. L_u , L_v and $L_w \in \Re$ are the scaling lengths defined as 492ft and $V \in \Re$ is the true airspeed of the quadrotor UAV. For a position hold simulation, V is equivalent to the external wind speed acting on the quadrotor UAV. The turbulence components added to the wind acceleration vector are determined by taking the time derivative of Equations (5.1), (5.2) and (5.3). $K_{turbulence\ severity} \in \Re$ is a constant applied to scale the output of the filter to alter the significance of the turbulence, as shown as a multiplying gain in Figure 5.8.

For this research, the most adverse wind conditions experienced by a UAV in flight are assumed to be with a wind speed of 8kts and a turbulence severity of 5. The standard deviation of the turbulence magnitude, where the turbulence magnitude is equal to the norm of a vector containing the outputs of Equations (5.1), (5.2) and (5.3), as a

function of wind speed and turbulence severity for a simulation time of 20 sec can be seen in Table 5.3. Table 5.3 shows that the spread of the turbulence magnitude increases as the wind speed and turbulence severity increase.

Table 5.4 shows the turbulence magnitude as a percentage of the wind speed. This table gives an indication of the degree of contribution of the turbulence to the total wind disturbance velocity magnitude. Table 5.4 shows that roughly 2% of the total wind disturbance originates from the wind turbulence as modeled by Equations (5.1), (5.2) and (5.3).

Table 5.3

Standard Deviation of Turbulence Magnitude as a Function of Wind Speed and Turbulence Severity

Turbulence Severity → Wind Speed (kts) ↓	1	2	3	4	5
1	0.0085	0.0170	0.0255	0.0340	0.0425
2	0.0118	0.0235	0.0353	0.0470	0.0588
3	0.0141	0.0282	0.0423	0.0564	0.0705
4	0.0160	0.0319	0.0479	0.0639	0.0798
5	0.0175	0.0350	0.0526	0.0701	0.0876
6	0.0189	0.0377	0.0566	0.0754	0.0943
7	0.0200	0.0400	0.0601	0.0801	0.1001
8	0.0211	0.0421	0.0632	0.0842	0.1053

Figure 5.9 shows an example wind gust magnitude for a 60 sec simulation with wind speed of 6kts, turbulence severity equal to 3 with the wind direction equal to 30 degrees from North. The blue line represents the output of the wind block as shown in Figure 5.8. There was observed that if the wind frequency was too high, the effects of the

wind disturbance on the simulated dynamics of the UAV could not be seen. The red line therefore shows the sampled wind speed used in the simulation.

Table 5.4

Turbulence Magnitude as a Percentage of the Wind Speed

Turbulence Severity → Wind Speed (kts) ↓	1	2	3	4	5
1	1.65%	3.30%	4.96%	6.61%	8.26%
2	1.15%	2.28%	3.43%	4.57%	5.72%
3	0.91%	1.83%	2.74%	3.65%	4.57%
4	0.78%	1.55%	2.33%	3.11%	3.88%
5	0.68%	1.36%	2.05%	2.73%	3.41%
6	0.61%	1.22%	1.83%	2.44%	3.06%
7	0.56%	1.11%	1.67%	2.22%	2.78%
8	0.51%	1.02%	1.54%	2.05%	2.56%

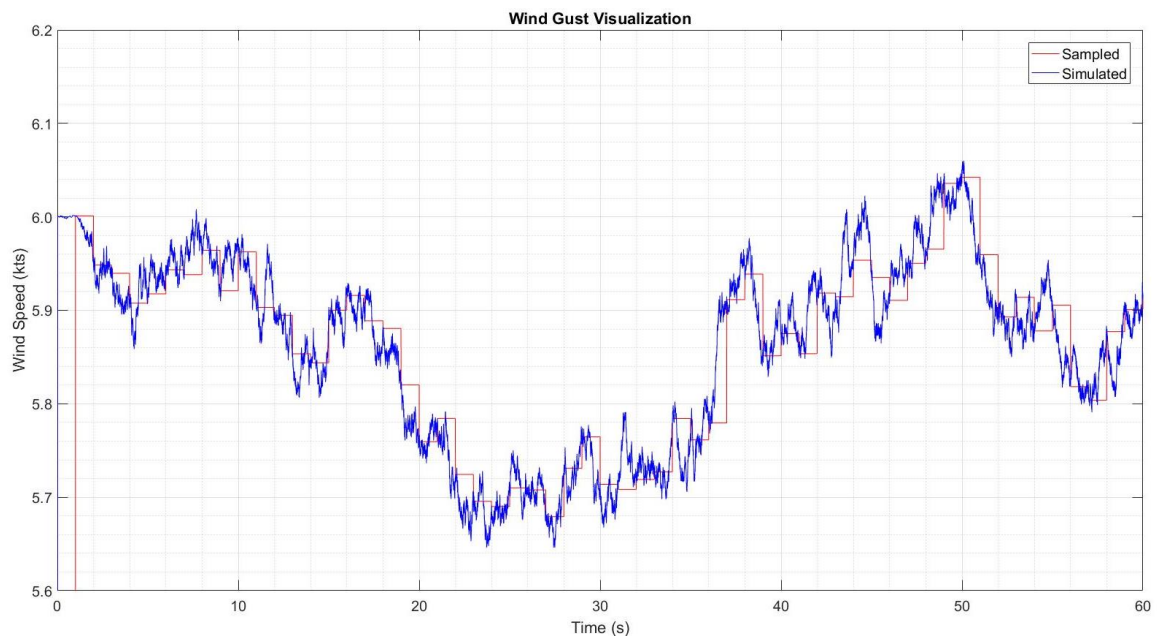


Figure 5.9. Wind speed magnitude time history with wind speed 6kts, turbulence severity 3 and wind direction 30deg from North.

5.2. Performance Metric

A performance metric is required to objectively evaluate and compare the performance of the developed control algorithms under wind disturbance conditions in both simulation and implementation. The performance metric must include relevant states required to evaluate the stability and tracking performance of the quadrotor's rotational and translational dynamics when the control algorithms are applied.

The performance metric combines the accumulated error between the commanded and measured values for translational displacement tracking, vertical displacement tracking, attitude angles tracking and angular rates tracking (Rivera K. , 2018). Since the available energy of the quadrotor UAV is limited, it is important to include the energy spent resulting from the application of the control algorithms in the performance metric evaluation. In order to evaluate the energy consumed by the system, an accumulation of the pulse-width-modulation signal (PWM) is included.

The trajectory tracking performance metric, which evaluates the tracking performance of translational and vertical displacement commands by the quadrotor UAV can be defined as:

$$\begin{aligned} \tilde{e}_{TT} = & \frac{1}{3} \left(1 - \frac{1}{C_X} \sqrt{\frac{1}{T} \int_0^T e_X^2 dt} \right) + \frac{1}{3} \left(1 - \frac{1}{C_Y} \sqrt{\frac{1}{T} \int_0^T e_Y^2 dt} \right) \\ & + \frac{1}{3} \left(1 - \frac{1}{C_Z} \sqrt{\frac{1}{T} \int_0^T e_Z^2 dt} \right) \end{aligned} \quad (5.4)$$

where C_X , C_Y and $C_Z \in \mathfrak{R}$ are normalization factors based on the worst case performance obtained in a set of tests, e_X , e_Y and $e_Z \in \mathfrak{R}$ are the error signals obtained by subtracting

the commanded values from the measured values for the navigational commands in the inertial reference frame, and $T \in \mathfrak{R}$ is the time duration of the specific test. As it can be seen in Equation (5.4), the performance metric for translational and vertical tracking performance can be expressed as the weighted sum of the magnitude and time normalized error accumulation for each one of the three displacements. Since in the simulation and implementation environment an autonomous position and altitude hold will be commanded, this trajectory tracking performance metric effectively becomes a position and altitude hold performance evaluation.

In a similar fashion, the attitude tracking performance metric can be defined as:

$$\begin{aligned} \tilde{e}_\theta = & \frac{1}{3} \left(1 - \frac{1}{C_\theta} \sqrt{\frac{1}{T} \int_0^T e_\theta^2 dt} \right) + \frac{1}{3} \left(1 - \frac{1}{C_\phi} \sqrt{\frac{1}{T} \int_0^T e_\phi^2 dt} \right) \\ & + \frac{1}{3} \left(1 - \frac{1}{C_\psi} \sqrt{\frac{1}{T} \int_0^T e_\psi^2 dt} \right) \end{aligned} \quad (5.5)$$

where C_θ , C_ϕ and $C_\psi \in \mathfrak{R}$ are normalization factors based on the worst case performance obtained in a set of tests, e_θ , e_ϕ and $e_\psi \in \mathfrak{R}$ are the error signals obtained by subtracting the commanded values from the measured values for the attitude angles.

The angular rates tracking performance metric is defined in Equation (5.6) where C_p , C_q and $C_r \in \mathfrak{R}$ are normalization factors based on the worst case performance obtained in a set of tests, e_p , e_q and $e_r \in \mathfrak{R}$ are the error signals obtained by subtracting the commanded values from the measured values for the angular rates:

$$\begin{aligned} \tilde{e}_\Omega = & \frac{1}{3} \left(1 - \frac{1}{C_p} \sqrt{\frac{1}{T} \int_0^T e_p^2 dt} \right) + \frac{1}{3} \left(1 - \frac{1}{C_q} \sqrt{\frac{1}{T} \int_0^T e_q^2 dt} \right) \\ & + \frac{1}{3} \left(1 - \frac{1}{C_r} \sqrt{\frac{1}{T} \int_0^T e_r^2 dt} \right) \end{aligned} \quad (5.6)$$

The accumulation of the PWMs activation can be obtained by summing the commanded PWM signals for each one of the four motors to characterize the energy consumed by the system:

$$\tilde{e}_{PWM} = 1 - \frac{1}{C_{PWM}} \sum_{i=1}^4 \left(\sqrt{\frac{1}{T} \int_0^T PWM_i dt} \right) \quad (5.7)$$

where $C_{PWM} \in \Re$ is a normalization factor based on the worst case performance obtained in a set of tests, $PWM_i \in \Re$ is the PWM signal commanded for motor i .

The weighted combination of these four individual metrics will yield the global performance index (PI), which characterizes the overall performance of the quadrotor UAV in a specific configuration:

$$PI = w_1 \tilde{e}_{TT} + w_2 \tilde{e}_\theta + w_3 \tilde{e}_\Omega + w_4 \tilde{e}_{PWM} \quad (5.8)$$

where w_1, w_2, w_3 and $w_4 \in \Re$ are arbitrary weights chosen based on the importance of each individual metrics. The arbitrary weights are case dependent and will therefore be presented in Chapters 5 and 7 in combination with the simulation and flight test results.

A good performance by the quadrotor UAV is characterized by a value close to one with one being a perfect tracking of values with no error for the metrics defined in

Equations (5.4), (5.5) and (5.6). Since there is no such thing as zero energy consumption for quadrotor UAV systems in flight, a value close to one for the metrics in Equation (5.7) will indicate a good economic energy performance. The overall PI defined in Equation (5.8) will show close to one for well performing quadrotor UAV systems under external wind disturbance.

The above presented performance metric was implemented in the simulation environment in the ‘Performance Metric’ block in Figure 5.1.

5.3. Performance Evaluation Baseline NLDI under Nominal Conditions

Simulation results can be obtained for the baseline NLDI under no wind conditions. This will establish the baseline NLDI’s performance under nominal conditions to which all other simulations results can be compared.

The nominal baseline NLDI evaluation consists of a 60 sec position and altitude hold at waypoint $(X_d, Y_d, H_d) = (0, 0, 10)\text{ft}$ for initial condition $(X_0, Y_0, H_0) = (0, 0, 10)\text{ft}$ with all initial rotational velocities, rotational displacements and translational velocities zero. The simulation is performed at a simulation frequency of $T_s = 500\text{Hz}$ for no wind conditions, meaning $\dot{W}_u = \dot{W}_v = \dot{W}_w = W_u = W_v = W_w = 0$ and all wind induced forces and moments described in Sections 3.2.2 and 3.2.3 are ignored.

Figure 5.10, Figure 5.11 and Figure 5.12 show the tracking performance for the attitude angles, angular rates and translational and vertical displacements. As expected, the baseline NLDI performs near perfect in a simulation environment free from external disturbances.

This nominal simulation evaluation can be repeated for the extended NLDI, baseline NLDI with adaptive ANN augmentation and baseline NLDI with \mathcal{L}_1 output-

feedback adaptive control augmentation. It can be shown that the adaptive controllers do not generate an augmented value since no uncertainties and/or disturbances are present in the environment. This is verified in Table 5.5, which shows the performance index for the baseline NLDI, extended NLDI, baseline NLDI with adaptive ANN augmentation and baseline NLDI with \mathcal{L}_1 output-feedback adaptive control augmentation under nominal conditions in the simulation environment where $w_1 = w_2 = w_3 = 0.3$ and $w_4 = 0.1$ as defined in Equation (5.8).

The values for the performance metric obtained for the four controllers under nominal conditions can be characterized as the best obtainable results in this architecture to which the further obtained simulation results can be compared.

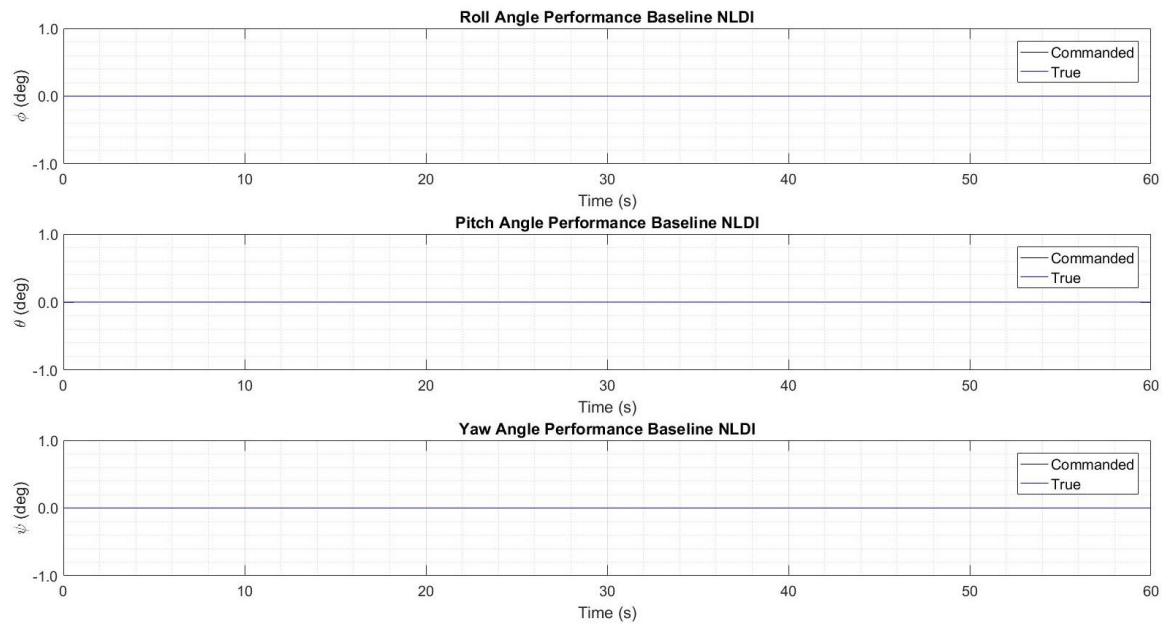


Figure 5.10. Attitude angles tracking performance baseline NLDI under nominal conditions.

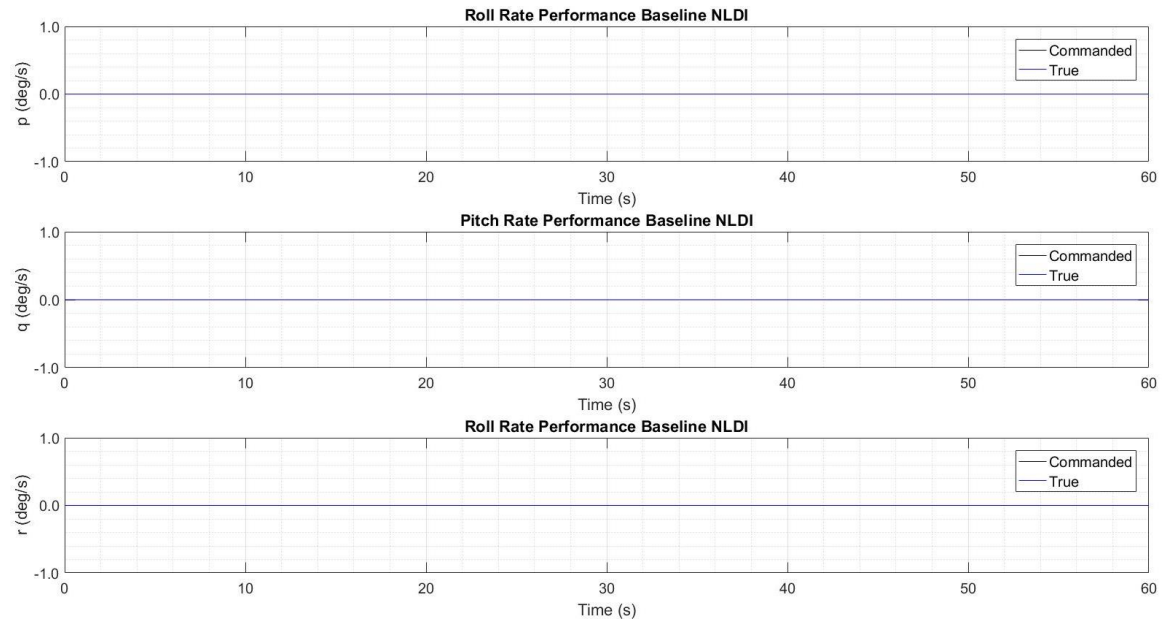


Figure 5.11. Angular rates tracking performance baseline NLDI under nominal conditions.

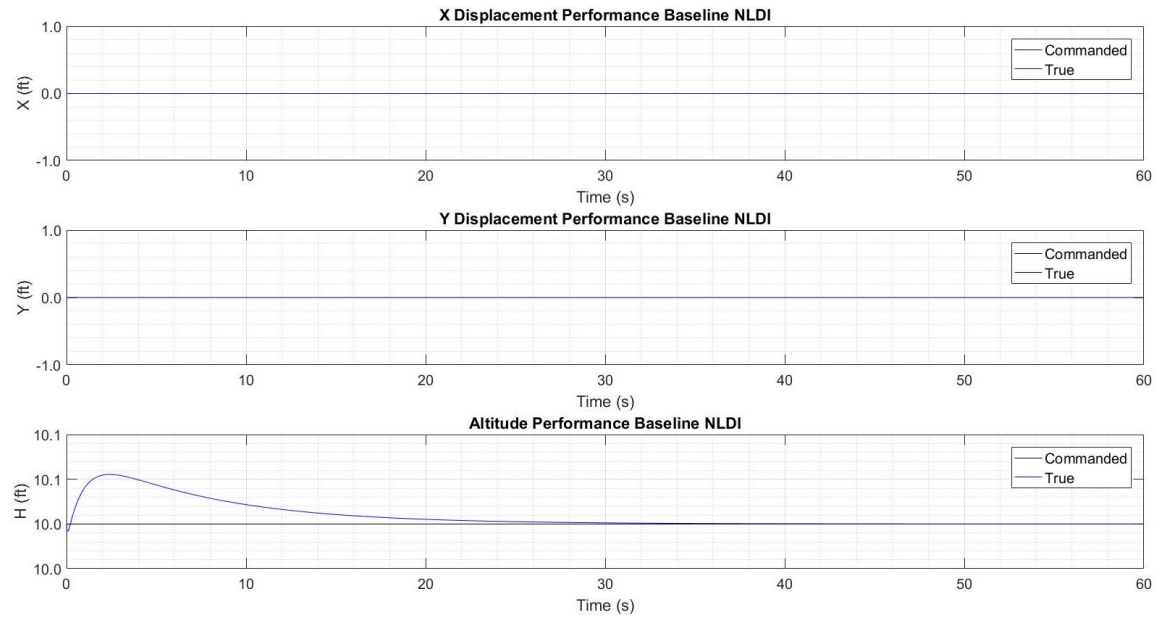


Figure 5.12. Translational and vertical tracking performance baseline NLDI under nominal conditions.

Table 5.5

Performance Index Comparison under Nominal Conditions in Simulation

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI
Baseline NLDI	1.0000	1.0000	0.9762	0.333750	0.9262
Extended NLDI	1.0000	1.0000	0.9762	0.333750	0.9262
NLDI + NN	0.9999	0.9999	0.9761	0.333750	0.9262
NLDI + L1	1.0000	1.0000	0.9762	0.333750	0.9262

5.4. Performance Evaluation Case Study

Now that the nominal condition is established and characterized, the discussed control algorithms in Chapter 4 can be evaluated under external wind disturbance in a simulation environment. Before considering the Monte Carlo analysis, which will provide a performance evaluation over the entire selected wind envelope, a set wind condition can be selected for a detailed analysis to provide some insight into the individual control algorithms. A randomly chosen external wind disturbance scenario characterized by a wind speed of 6kts headed towards 30 degrees North with a turbulence severity of 3 (Table 5.3 and Table 5.4) is selected for this analysis. The simulation will once again consist of a 60 sec position and altitude hold at waypoint $(X_d, Y_d, H_d) = (0, 0, 10)\text{ft}$ for initial condition $(X_0, Y_0, H_0) = (0, 0, 10)\text{ft}$ with all initial rotational velocities, rotational displacements and translational velocities zero. The simulation is performed at a simulation frequency of 500Hz.

Figure 5.13 and Figure 5.14 show a top view and 3D view of the position hold comparison performed by the four controllers at the stated wind condition. As it is shown, the extended NLDI controller outperforms the other controllers in the position and

altitude hold capabilities since it recovers the quadrotor to the desired location initially and maintains a compact trajectory in both position and altitude throughout the remainder of the simulation. A steady-state error in the extended NLDI response can be seen indicating that the extended NLDI is not able to perfectly invert and cancel the wind disturbance. This is caused by the virtual controllers, which enforce transient dynamics containing inherent delays on the quadrotor response, making it impossible for the extended NLDI to perfectly cancel the wind disturbance. Another factor that prevents a perfect cancelation is the inability to invert and correct for the stochastic wind turbulence. The baseline NLDI, baseline NLDI with adaptive ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation perform similarly with respect to the position and altitude hold capabilities.

The performance improvement of the NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation with respect to the baseline NLDI is not quite distinguishable from Figure 5.13 and Figure 5.14, but can be seen in the performance index in Table 5.6 for $w_1 = w_2 = w_3 = 0.3$ and $w_4 = 0.1$. The extended NLDI outperforms the other controllers at position hold capabilities, while the adaptive controllers increase the overall rotational stability of the quadrotor UAV system in comparison to the baseline NLDI control law. As expected, the NLDI with ANN augmentation provides the best performance in angular rates tracking since the angular rates tracking error drives the ANN updating. A similar result is presented for the NLDI with \mathcal{L}_1 augmentation, which promotes a better performance in attitude angle tracking due to the inherent design focusing on the attitude angles tracking error. Clearly, the stability of the quadrotor is increased with the application of the adaptive ANN and \mathcal{L}_1 augmentation. The

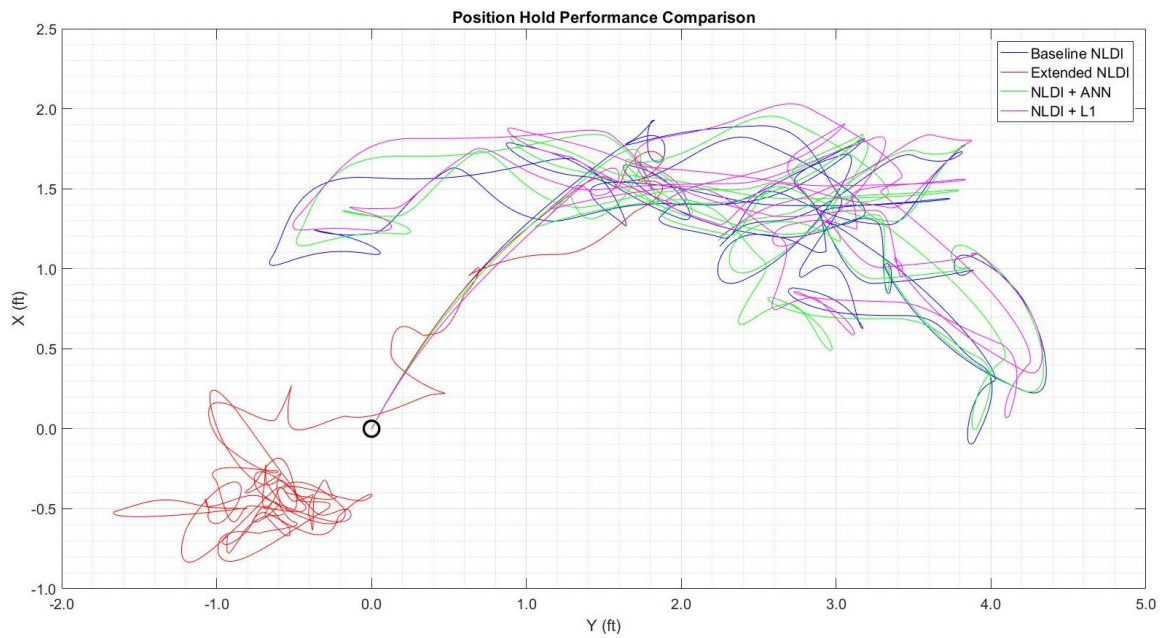


Figure 5.13. Top view 3D simulated position hold performance comparison.

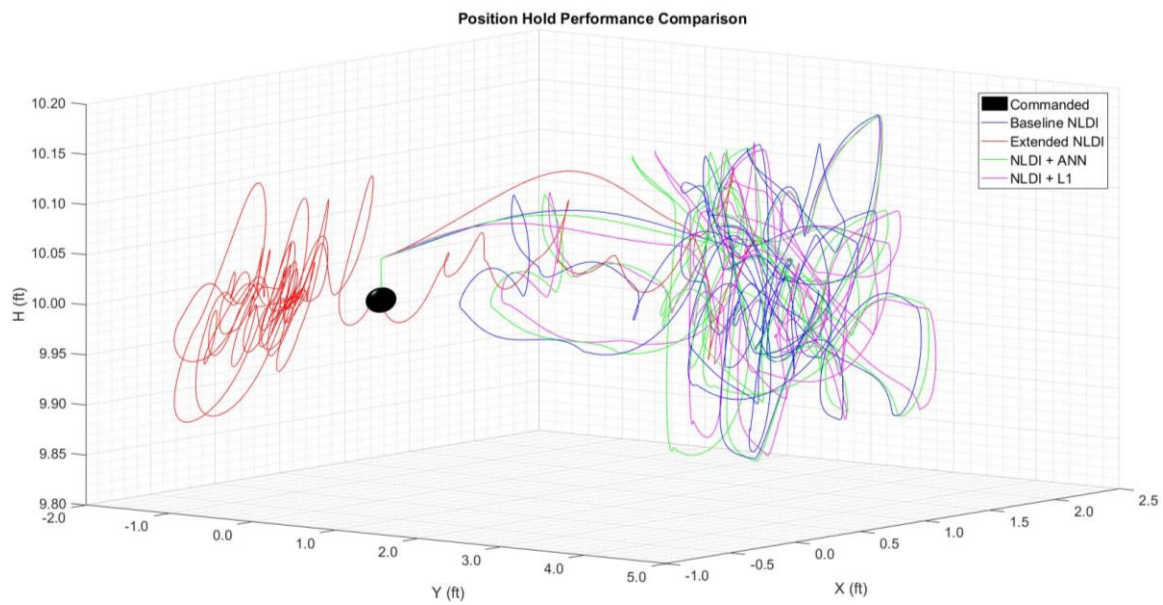


Figure 5.14. 3D simulated position hold performance comparison.

performance metric with respect to energy consumption is fairly constant across the board with the baseline NLDI with \mathcal{L}_1 adaptive control performing slightly better.

Overall, the NLDI with ANN augmentation results in the highest PI, indicating the best performance in comparison with the other methods.

The extended NLDI, baseline NLDI with adaptive ANN augmentation and NLDI with \mathcal{L}_1 output-feedback adaptive control augmentation can be analyzed in further detail to create a better understanding of the overall performance.

Table 5.6

Performance Index for Performance Comparison

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI
Baseline NLDI	0.9212	0.9361	0.7581	0.333807	0.8180
Extended NLDI	0.8596	0.8758	0.8925	0.333786	0.8218
NLDI + NN	0.9328	0.9558	0.7592	0.333816	0.8277
NLDI + L1	0.9529	0.9282	0.7565	0.333842	0.8246

5.4.1. Extended NLDI. Figure 5.15 and Figure 5.16 present a comparison of the time histories for attitude angles and angular rates for the performed simulation, which show that the application of the extended NLDI results in larger magnitude angles and rates in the quadrotor UAV. This is especially true in the pitch axis since this is the primary axis the wind is acting in. These high magnitude angles and rates are caused by the direct inclusion of wind accelerations in the outer loop to generate commanded pitch and roll angles as described in Equations (4.40) and (4.41), and by the direct inclusion of wind induced moments to generate a control signal commanded to the control allocation of the UAV control laws as seen in Equation (4.24). The filtering of wind velocities and accelerations would increase the stability of the rotational dynamics of the quadrotor

UAV equipped with the extended NLDI, but inherent information of the wind would be lost in this process. This would degrade the position hold capabilities of the extended NLDI. The extended NLDI therefore shows a trade-off between rotational and translational tracking stability.

5.4.2. Adaptive ANN augmentation. In order to appreciate the augmentation by the adaptive ANN, it is first necessary to show the behavior of the baseline NLDI fast loop response discussed in Section 4.1.1. Figure 5.17 shows a time history of the virtual control inputs under wind disturbance conditions in simulation for the baseline NLDI, which was first introduced in Equation (4.2). As can be seen, the virtual control input takes the shape of an offset value in the roll and pitch rate channels caused by the wind disturbance.

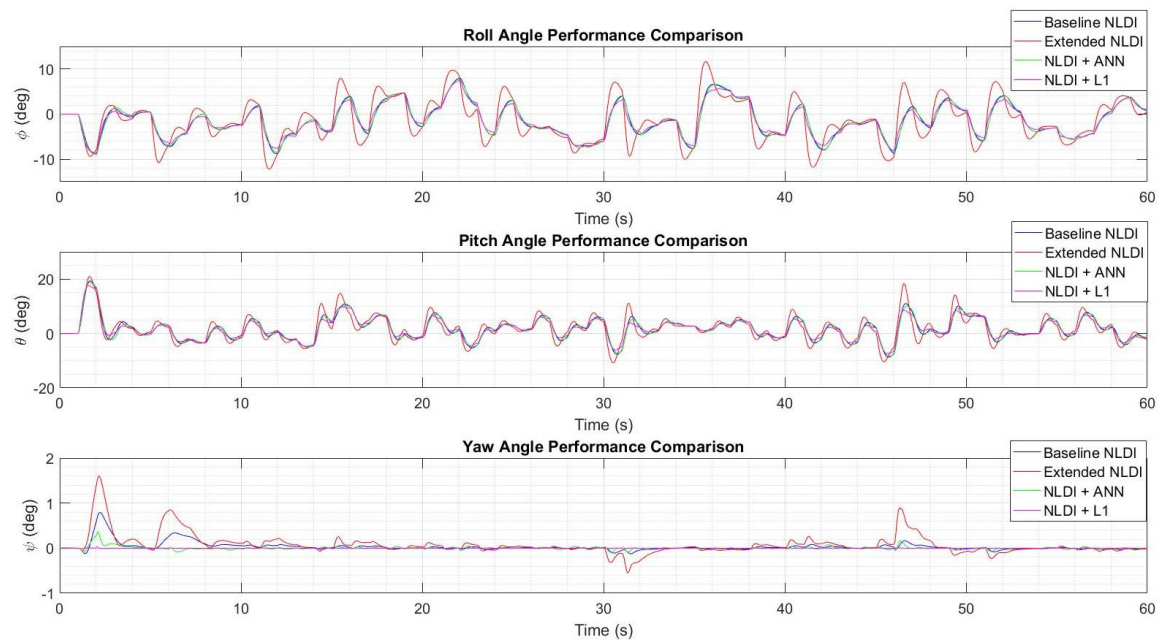


Figure 5.15. Attitude angles time history comparison under wind disturbance conditions in simulation.

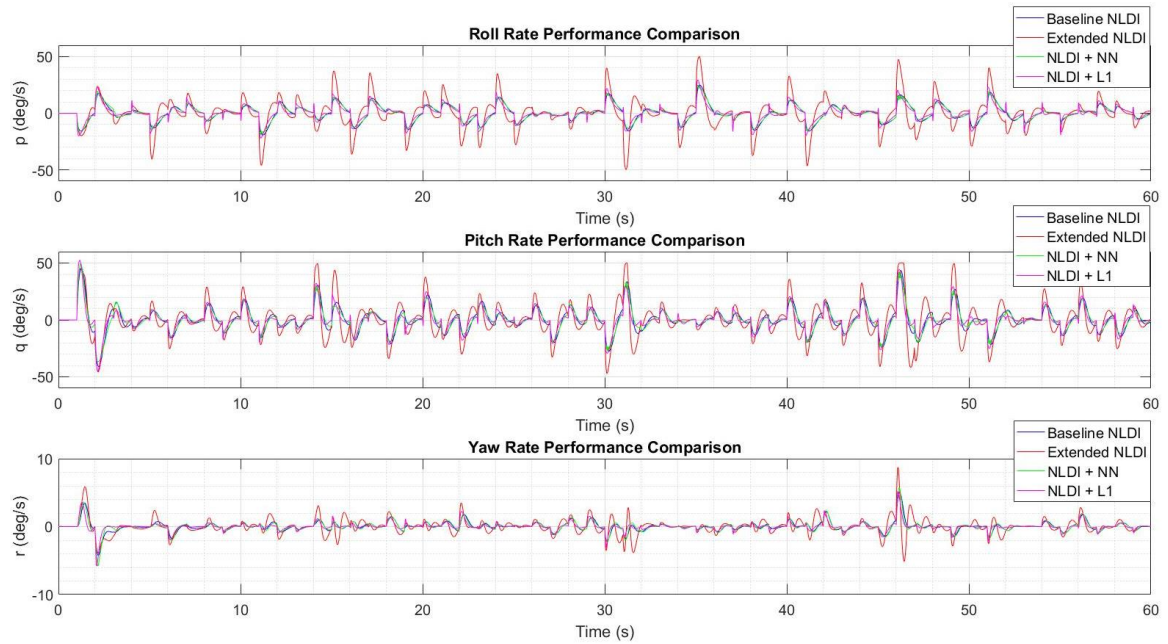


Figure 5.16. Angular rates time history comparison under wind disturbance conditions in simulation.

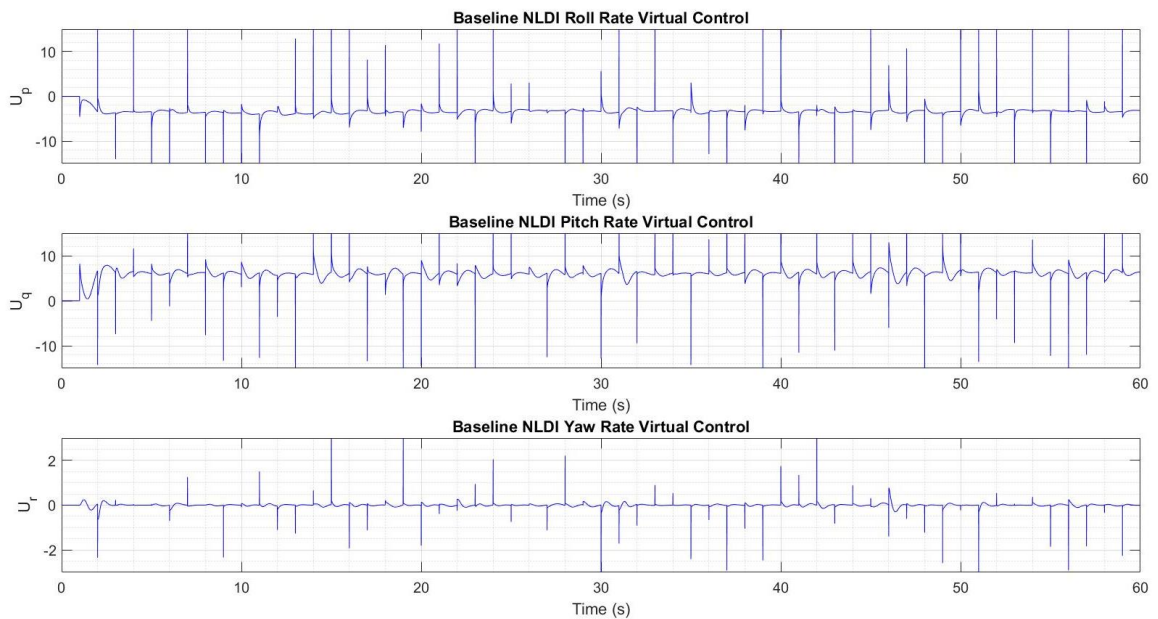


Figure 5.17. Baseline NLDI virtual control under wind disturbance conditions in simulation.

Figure 5.18 shows the outputs of the adaptive ANN under the considered wind conditions as defined in Equation (4.53). Figure 5.19 shows the virtual control signal generated by the NLDI with ANN augmentation in all three channels as defined in Equation (4.58). As it can be seen, the adaptive ANN actively augment a control signal in the roll and pitch channels, U_{pNN} and U_{qNN} , in an effort to aid the tracking of the desired roll and pitch rates commanded by the baseline NLDI control law; U_{pNLDI} and U_{qNLDI} . More activation in the pitch channel seems to be present, which is expected since the wind disturbance direction is towards 30 degrees North causing the largest uncertainties and inversion errors in the pitch channel. The output of the ANN for the roll and pitch channels, shows an augmented signal with a constant offset as shown in Figure 5.18. This signal drives the error signal sent to the ANN, U_{pNLDI} and U_{qNLDI} back to zero, which inherently shows a constant offset as presented in Figure 5.17. The output of the ANN for the roll and pitch channels shows high frequency corrections on top of the constant offset. This high frequency signal is generated to correct for the high frequency wind turbulence. The ANN in the yaw channel does not activate at the same intensity as the ANN applied in the roll and pitch channels, indicating that the yaw inversion does not contain much uncertainty under the simulated wind disturbance.

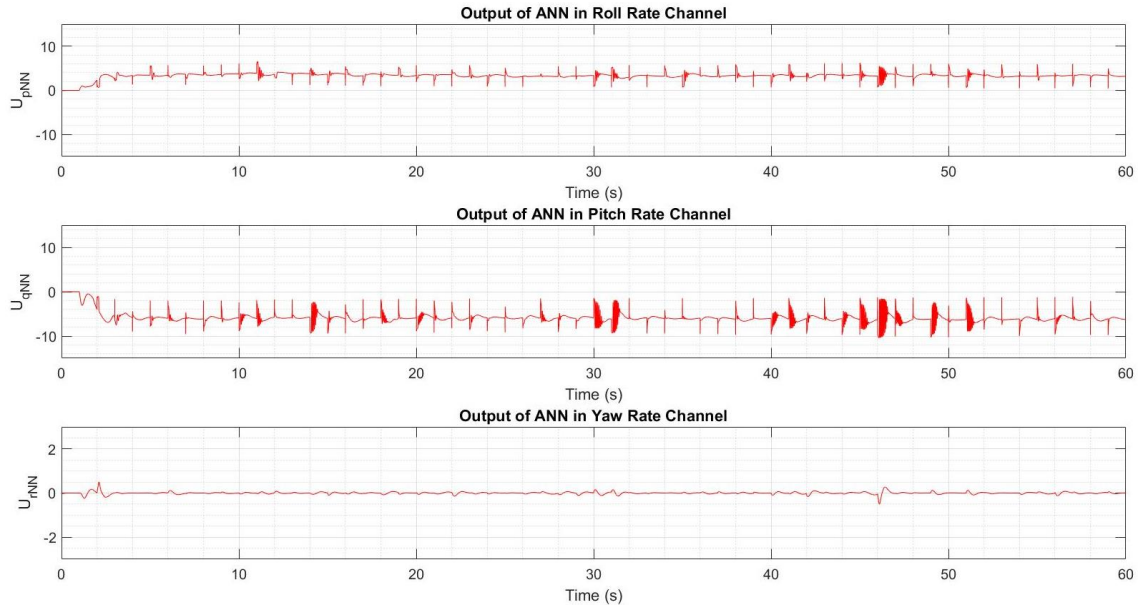


Figure 5.18. Outputs of the adaptive ANN under wind disturbance conditions in simulation.

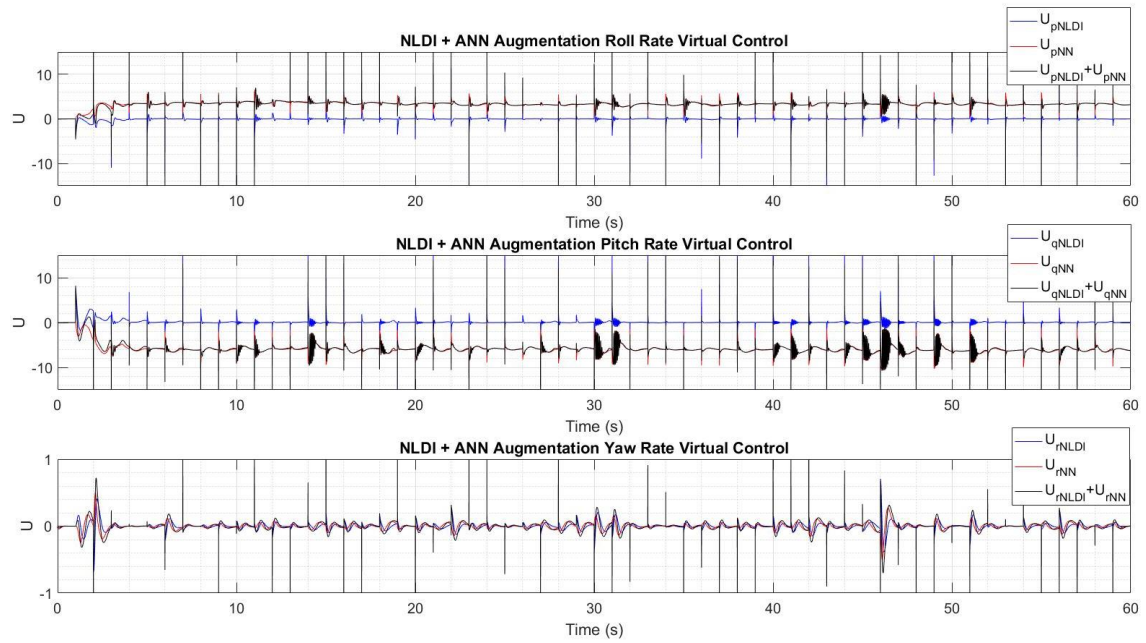


Figure 5.19. NLDI with adaptive ANN augmentation virtual control under wind disturbance conditions in simulation.

5.4.3. \mathcal{L}_1 output-feedback adaptive control augmentation. Figure 5.20

shows the outputs of the \mathcal{L}_1 adaptive controller under the simulated wind conditions as defined in Equation (4.78). Figure 5.21 shows the virtual control signal generated by the NLDI with the \mathcal{L}_1 adaptive control augmentation for all three channels as defined in Equation (4.89). It is shown that the \mathcal{L}_1 adaptive controller actively augments the roll and pitch rate channels, where the pitch channel produces a larger augmented value. The augmented values in the roll and pitch channel, U_{pL1} and U_{qL1} , are characterized by a similar magnitude and frequency as the virtual roll and pitch rate control coming from the baseline NLDI virtual controller; U_{pNLDI} and U_{qNLDI} .

Comparing the ANN augmentation and \mathcal{L}_1 adaptive augmentation performance, it can be shown in Figure 5.18 and Figure 5.20 that both controllers attempt to aid the baseline NLDI by producing an augmented correction signal. Both controllers augment an offset value where the ANN outputs also show high frequency adaptation for turbulence.

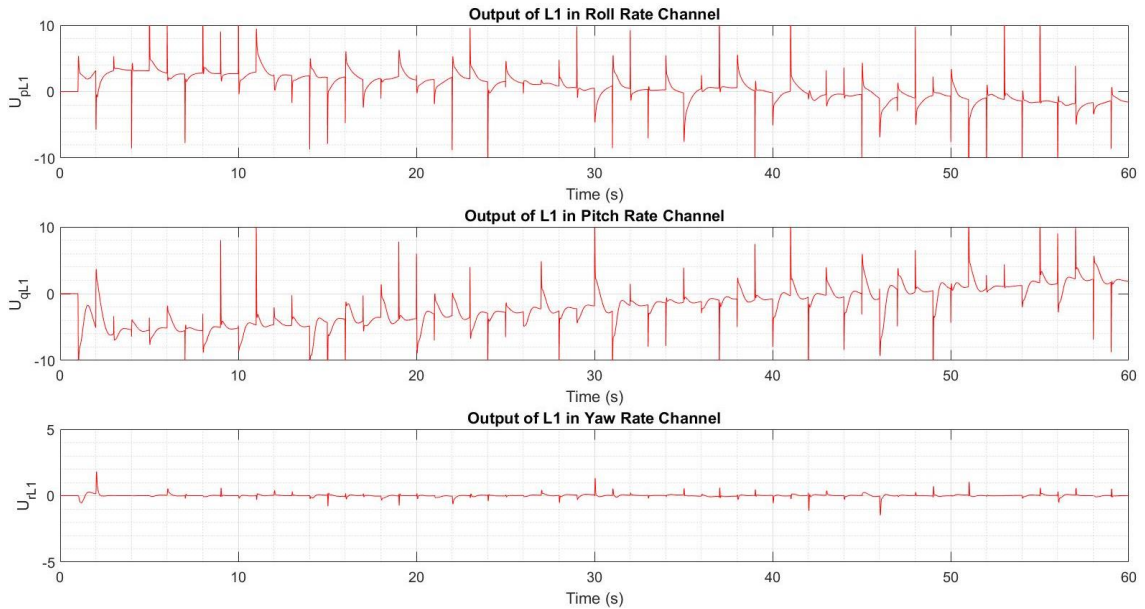


Figure 5.20. Outputs of the \mathcal{L}_1 adaptive controller for nominal indoor flight testing under wind disturbance conditions in simulation.

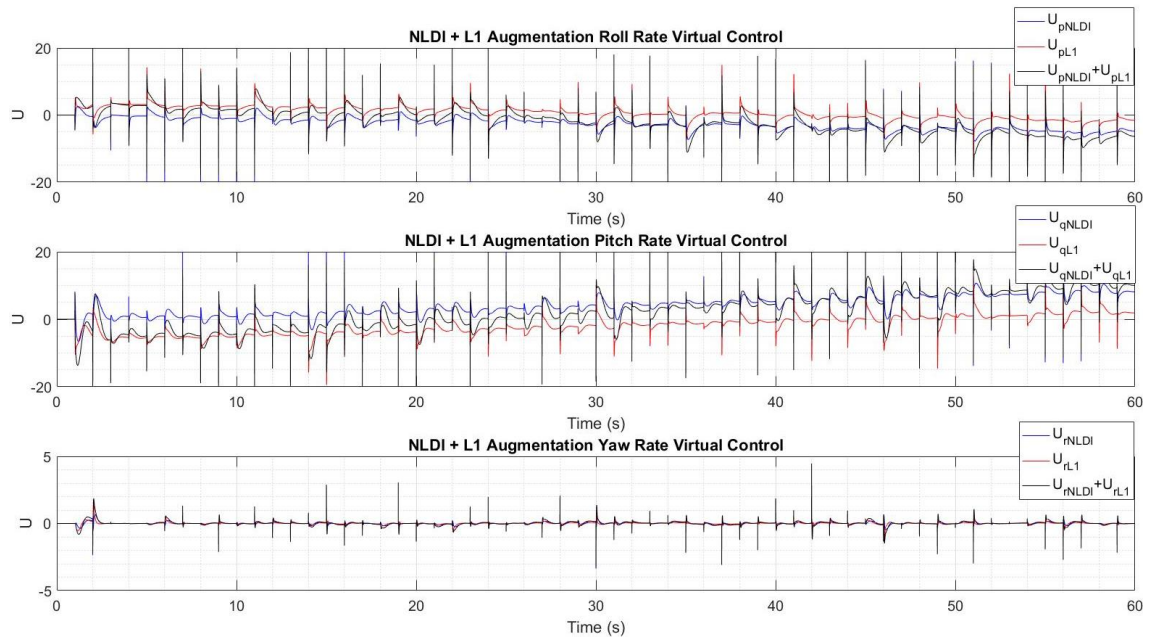


Figure 5.21. NLDI with \mathcal{L}_1 adaptive controller augmentation virtual control under wind disturbance conditions in simulation.

5.5. Monte Carlo Analysis

Section 5.4 provides a comparison analysis for the developed control algorithms at one wind condition. A Monte Carlo analysis was applied to evaluate and compare the performance of the four control algorithms throughout the entire considered wind envelope. As previously mentioned, a realistic most adverse wind condition experienced by a quadrotor UAV in flight can be set at a wind speed magnitude of 8kts with a turbulence severity of 5. Additionally, the wind direction can be varied between 0 and 359 degrees from North. In order to evaluate the performance of the control algorithms throughout this wind envelope, ten equally spaced wind speed magnitudes between 0 and 8kts, ten equally spaced turbulence severity values between 0 and 5, and five wind directions between 0 and 359 degrees were sampled as outlined in Table 5.7

The four control approaches were evaluated at each possible combination of wind

speed magnitude, wind turbulence severity and wind direction resulting in 500 evaluation points. At each evaluation point, the performance of the controllers was evaluated with a 20 sec position and altitude hold at waypoint $(X_d, Y_d, H_d) = (0, 0, 10)$ ft for initial condition $(X_0, Y_0, H_0) = (0, 0, 10)$ ft with all initial rotational velocities, rotational displacements and translational velocities zero. The simulations were run at a frequency of 500Hz.

Table 5.7

Wind Condition Sample Points for Monte Carlo Analysis

Wind Magnitude (kts)	0.00	0.89	1.78	2.67	3.56	4.44	5.33	6.22	7.11	8.00
Wind Turbulence Severity	0.00	0.56	1.11	1.67	2.22	2.78	3.33	3.89	4.44	5.00
Wind Direction from North (deg)	0	72	144	216	288					

The results of the Monte Carlo analysis are presented by plotting the performance metric development for attitude tracking, angular rates tracking, trajectory tracking and PWM accumulation as a function of wind speed magnitude and turbulence severity at a set wind direction. A same plot can be generated for the global PI development with $w_1 = w_2 = w_3 = 0.3$ and $w_4 = 0.1$.

Figure 5.22 through Figure 5.26 show the development of the performance metrics for attitude angles, angular rates, position hold, energy consumption and global PI for increasing wind speed magnitude and turbulence severity at a wind direction of 288 degrees from North. The results show a similar trend as found in Section 5.4; the NLDI with \mathcal{L}_1 augmentation performs best in the attitude angle tracking, NLDI with ANN augmentation performs best in angular rate tracking and the extended NLDI performs best in position hold. The differences for energy consumption are very small

with the extended NLDI performing slightly better than the other three controllers. Figure 5.26 shows that the NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation result in the best performance overall.

Based on Figure 5.22 through Figure 5.26, it is shown that although the extended NLDI results in the best trajectory tracking performance, it does not necessarily produce the best stability characteristics as indicated by Figure 5.22 and Figure 5.23. This could be visualized by the quadrotor UAV with extended NLDI holding the desired position and altitude well but with attitude oscillations, performing even worse than the baseline NLDI. The NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation outperform the baseline NLDI in all plots, increasing the robustness of the quadrotor UAV under external wind disturbances.

The development of the performance metrics for attitude angles, angular rates, trajectory tracking, energy consumption and global PI for increasing wind speed magnitude and turbulence severity at a wind direction of 0, 72, 144 and 216 degrees from North can be found in the Appendix. A similar trend at the different wind direction conditions, as seen in Figure 5.22 through Figure 5.26, can be observed. The figures in the Appendix show that, for some wind speed magnitudes and turbulence severity combinations, the baseline NLDI and extended NLDI produce instabilities, as shown by a sharp decrease in the plotted plane. However, the augmentation by ANN results in the NLDI to be stable as compared to the baseline NLDI at these points. The instability is either delayed to a more adverse condition or stability is guaranteed throughout the envelope, which is not achievable with the baseline NLDI.

Figure A.11, Figure A.13, Figure A.14, Figure A.15, Figure A.16, Figure A.18

and Figure A.19 show an unexpected behavior by the NLDI with \mathcal{L}_1 augmentation. The NLDI with \mathcal{L}_1 augmentation goes unstable at a certain wind condition but is able to converge at a more adverse wind condition. These instabilities will be considered as outliers.

Table 5.8 shows the means for the global performance index and individual performance metrics over the 500 Monte Carlo runs. The means confirm the overall observed trend with the NLDI with adaptive ANN augmentation overall outperforming the other three controllers.

Table 5.9 summarizes standard deviation for the global performance index and individual metrics over the 500 Monte Carlo runs. The standard deviation information gives an indication of the consistency of the controller performance throughout the wind envelope. As it can be seen, the NLDI with adaptive ANN augmentation performs the most consistently in all runs, resulting in a smaller standard deviation than the other three controllers.

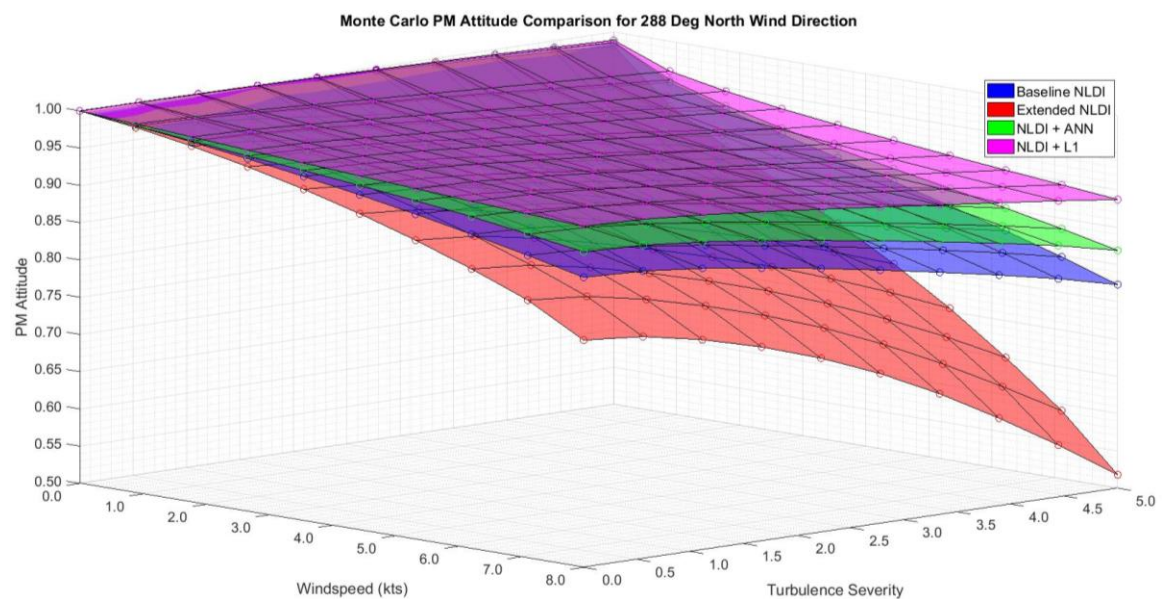


Figure 5.22. Controller comparison for attitude angles PM for wind 288 deg from North.

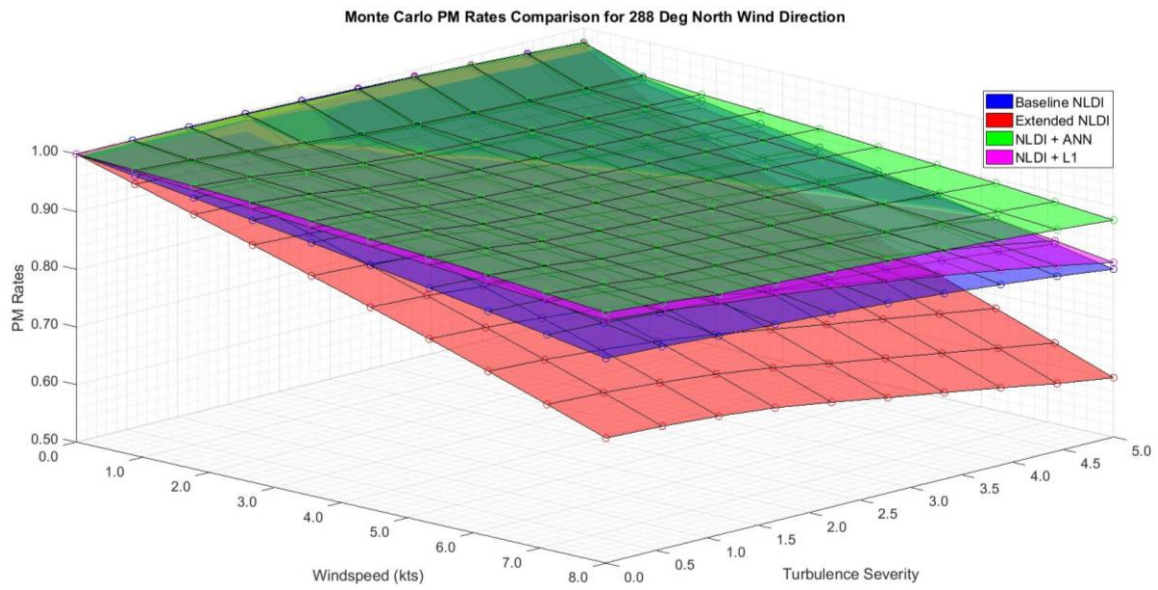


Figure 5.23. Controller comparison for angular rates PM for wind 288 deg from North.

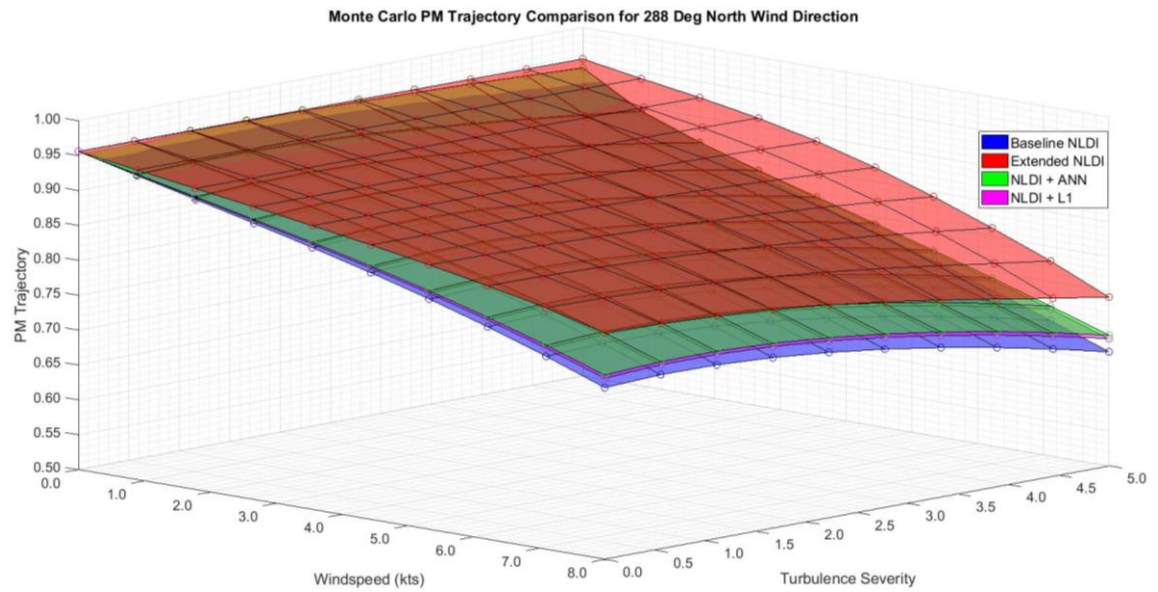


Figure 5.24. Controller comparison for trajectory PM for wind 288 deg from North.

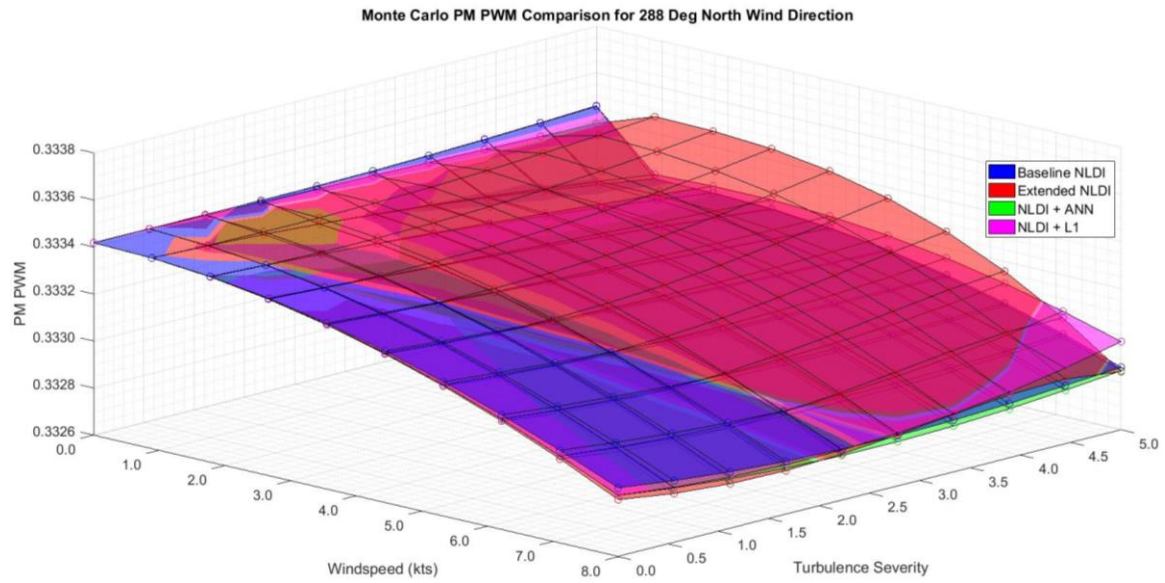


Figure 5.25. Controller comparison for PWM PM for wind 288 deg from North.

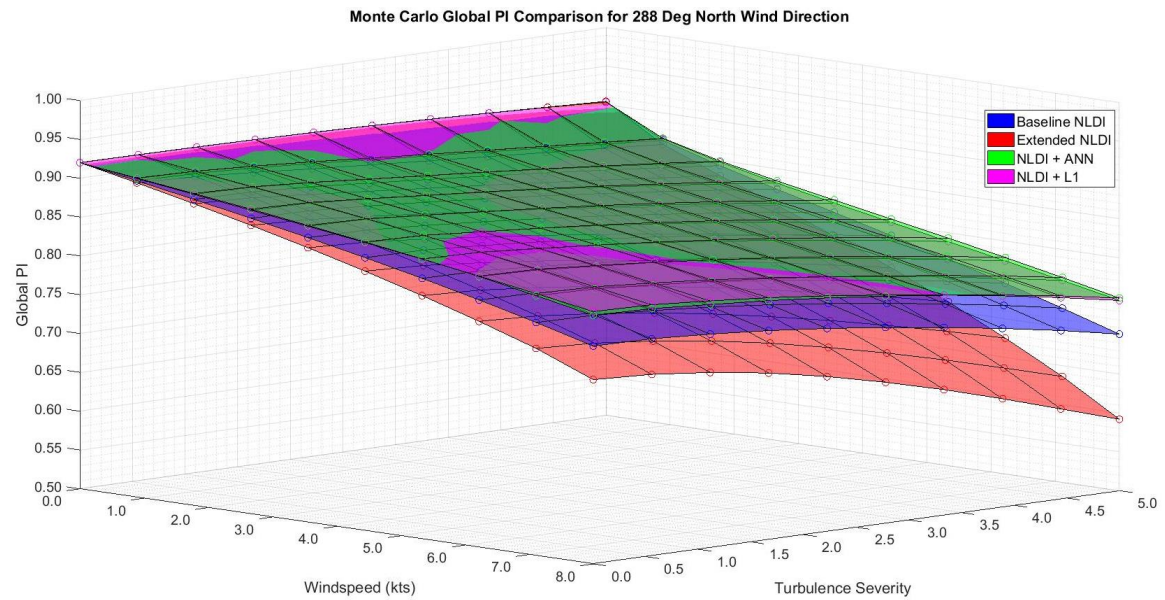


Figure 5.26. Controller comparison for global PI for wind 288 deg from North.

Table 5.8

Performance Metric Means for 500 Monte Carlo Runs

Controller	PM Trajectory Mean	PM Attitude Mean	PM Rates Mean	PM PWM Mean	Global PI Mean
Baseline NLDI	0.8165	0.9057	0.9002	0.33303	0.8199
Extended NLDI	0.8845	0.8320	0.8151	0.33310	0.7926
NLDI + ANN	0.8270	0.9288	0.9427	0.33322	0.8429
NLDI + L1	0.8222	0.9480	0.9023	0.33324	0.8333

Table 5.9

Performance Metric Standard Deviation for 500 Monte Carlo Runs

Controller	PM Trajectory SD	PM Attitude SD	PM Rates SD	PM PWM SD	Global PI SD
Baseline NLDI	0.1114	0.0875	0.0784	0.00318	0.0826
Extended NLDI	0.0951	0.1541	0.1276	0.00117	0.1111
NLDI + ANN	0.0947	0.0478	0.0289	0.00022	0.0502
NLDI + L1	0.1068	0.0526	0.1206	0.00021	0.0929

6. Quadrotor UAV Testbed & Implementation Environment

This chapter will introduce the UAV quadrotor research testbed, weather station and flight testing locations used to test the developed algorithms in flight. The hardware components consisting of the power and propulsion systems, sensors and Pixhawk flight computer are introduced first. This is followed by a discussion of the software implemented in the Pixhawk flight computer. This chapter will conclude with a presentation of the flight testing locations at which flight tests were performed and the weather station applied to characterize the wind in the outdoor environment.

6.1. DJI Flamewheel F330 Research Testbed

The research testbed used to evaluate the performance of the developed control algorithms in flight is the DJI Flamewheel F330 quadrotor. This quadrotor is symmetric about the body reference x- and y-axis with a crossed arm configuration. This low-cost testbed was chosen for its durability and ease of implementation of the required sensors and flight computer. Figure 6.1 shows the front/top view of the DJI Flamewheel F330 testbed with Pixhawk flight computer, GPS module with digital compass, brushless motors and eight-inch diameter propellers. Figure 6.2 shows the side view with the Lightware laser for altitude measurements, 30A ESCs and a 3000mAh 11.1V lithium-ion polymer (LiPo) battery required to power all systems.

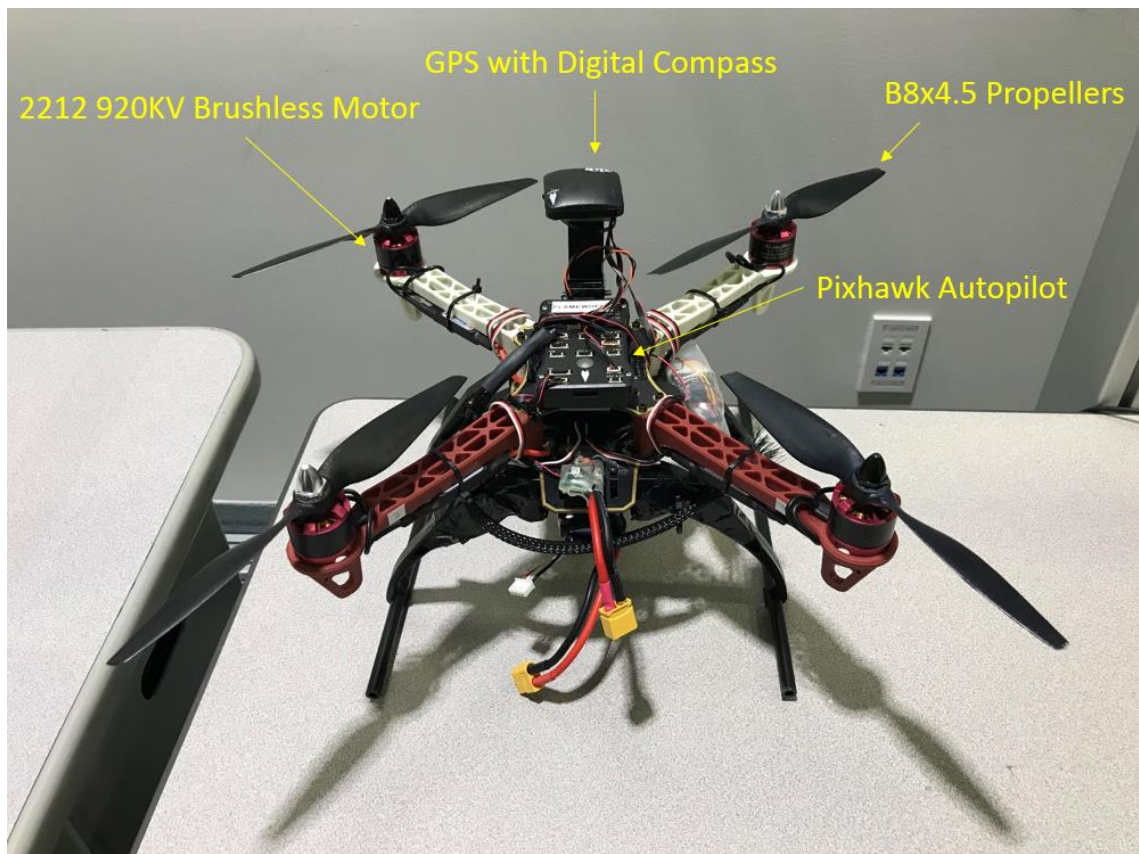


Figure 6.1. DJI Flamewheel F330 research testbed – front/top view.

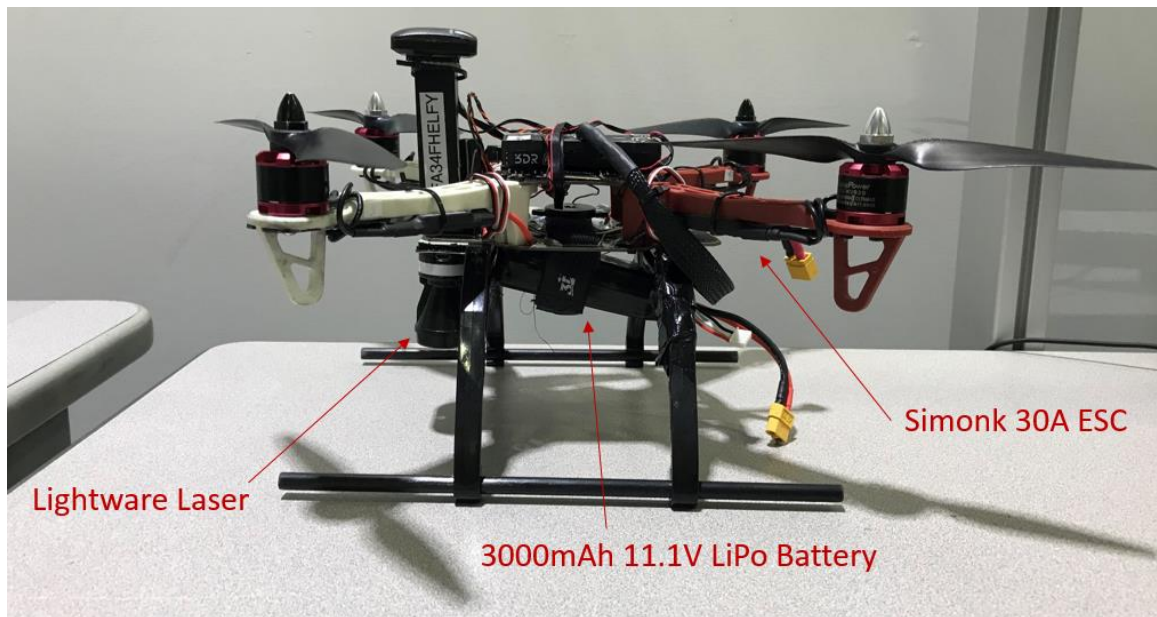


Figure 6.2. DJI Flamewheel F330 research testbed – side view.

Table 6.1 outlines the main properties of the DJI Flamewheel F330 research testbed with the heights measured from the landing gear base. The vertical offset of the motors with respect to the CG is 1inch.

Table 6.1

Main Properties of the DJI Flamewheel F330

Total Flight Weight	2.2lb
Thrust Arm	6.5in
CG Height	4in
Motor Height	5in
Max Height	8in
I_{xx}	$0.0132\text{slug}\cdot\text{ft}^2$
I_{yy}	$0.0132\text{slug}\cdot\text{ft}^2$
I_{zz}	$0.0230\text{slug}\cdot\text{ft}^2$

6.1.1. Pixhawk flight computer and sensor package. The primary flight computer equipped on the Flamewheel is the Pixhawk 1 (PX1) autopilot unit (Figure 6.3). Originally designed by 3D Robotics, PX1 offers internal processing at 180MHz with 256 KB RAM and 2MB of Flash. Although the processing capabilities of the PX1 are not outstanding, they are sufficient to run all required control laws including the adaptive controllers.

PX1 runs a soft real-time operating system (RTOS) where information about the current status of the board can be communicated by the light emitting diode (LED), multi-tone audio and data logging on the microSD card. PX1 includes 14 PWM outputs and a wide variety of additional connectivity options such as 12C, CAN, ADC and UART. The board is powered by the flight batteries.



Figure 6.3. PX1 autopilot board (Px4 Dev Team, 2019).

Various embedded sensors that are part of the standard autopilot unit, and external sensors that are required for navigation can be identified that are necessary for stable and controlled flight of the Flamewheel quadrotor.

MPU6000 3-axis accelerometer/gyroscope. The primary accelerometer and gyroscope used by the PX1 is the MPU6000 (Figure 6.4). This sensor provides 3-axis

angular rates and translational acceleration measurements in the body reference frame at a maximum frequency of 8kHz.

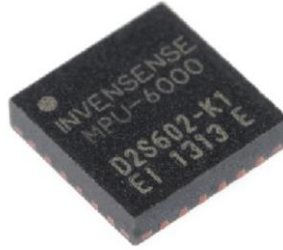


Figure 6.4. MPU6000 3-axis primary accelerometer/gyroscope (SparkFun Electronics, 2013).

ST micro L3GD20H 16-bit gyroscope. The L3GD20H 16-bit gyroscope is a secondary gyroscope sensor equipped in the PX1. It is defined as a low-power angular rate sensor which gives the possibility to communicate the measured angular rates at different bandwidths to external devices through the serial protocol I2C.

ST micro L3M303D 14-bit accelerometer/magnetometer. The L3M303D 14-bit accelerometer/magnetometer is a secondary accelerometer/magnetometer sensor equipped in the PX1. The L3M303D includes an I2C interface used to communicate readings through external connections.

UBLOX LEA-6H GPS receiver module with digital compass. The external UBLOX GPS includes the HMC5883L digital compass which provide the PX1 with a heading and a position measurement in the inertial reference frame. The GPS and digital compass are required to be connected to the PX1 through serial and I2C connection respectively. The inclusion of the digital compass in the external UBLOX GPS is convenient due to the possibility of mounting the compass away from potential

sources of interference. The GPS and digital compass offer a 5Hz update rate.



Figure 6.5. UBLOX LEA-6H GPS receiver module with digital compass (ArduPilot Dev Team, 2019).

Lightware SF11/C laser. The external Lightware SF11/C laser provides the PX1 with altitude readings throughout the flight (Figure 6.6). The laser is connected to the PX1 through serial connection offering altitude measurements with a range up to 120m.



Figure 6.6. Lightware SF11/C laser (ACRONAME, 2019).

DX8 8CH transmitter and DSMX remote receiver. The DX8 8CH transmitter is used during flight testing operations to allow for wireless communication with the PX1 onboard autopilot unit. The transmitter is equipped with eight channels which can be set up using four channels for primary flight controls, one channel for arming/unarming of the PX1 and three extra optional switches. The transmitter includes a

battery pack which offers about an hour of usage before recharge is required. The signals from the DX8 8CH transmitter are received by the DSMX remote receiver connected to the PX1. The DX8 8CH transmitter and DSMX remote receiver can be seen in Figure 6.7.



Figure 6.7. DX8 8CH transmitter and DSMX remote receiver (Horizon Hobby, 2019).

6.1.2. Power and propulsion systems. In order to provide power to the PX1 and to allow for the outputs of the PX1 to be send to the motors, various power and propulsion systems are required, which are introduced next.

2212 920KV brushless motors. The Flamewheel is equipped with four low-cost 2212 920KV brushless motors to provide the required propulsion (Figure 6.8). These high performance motors are easy to assembly and very durable. This motor is rated for 920 RPM/V (KV). Separate motors allow for clockwise and anticlockwise assembling of the propellers to distinguish the spin direction.



Figure 6.8. 2212 920KV brushless motors (Amazon, 2019).

Simonk 30A ESC. Low-cost Simonk 30A electronic speed controllers (ESCs) are applied to regulate the speed of the brushless motors. The ESCs will allow PWM commands from the PX1 to be converted to voltage input to the motors. These ESCs can be calibrated for the throttle range of the DX8 8XH transmitter for optimal performance. Additionally, the Simonk 30A offer various safety and protection modes.



Figure 6.9. Simonk 30A ESC (Amazon, 2019).

B8x4.5MR-B4 propellers. The Flamewheel is equipped with eight-inch diameter, 4.5 inch pitch propellers produced by Advanced Precision Composites (Figure 6.10). These cost-efficient plastic propellers are very durable and are compatible with the clockwise and anticlockwise rotation required for the quadrotor UAV.



Figure 6.10. B8x4.5MR-B4 propellers (Propellers, 2019).

HobbyStar 3000mAh 11.1V, 3S 30C LiPo battery. The HobbyStar 3000mAh 11.1V lithium-ion polymer (LiPo) battery is used to power the PX1 and propulsion systems of the Flamewheel quadrotor (Figure 6.11). This battery offers fast charging at a rate of 2C and powerful discharging capabilities at 30C, making it an ideal fit to power the Flamewheel.

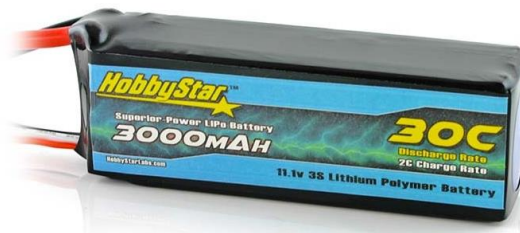


Figure 6.11. HobbyStar 3000mAh 11.1V LiPo battery (RCJuice, 2019).

6.2. Embedded Software

Embedded software is required to allow for the deployment of the developed control law architectures presented in Chapter 4 in the PX1 flight computer for inflight testing. MATLAB/Simulink provides embedded coder options to allow for the testing of MATLAB/Simulink generated codes with hardware-in-the-loop. One example is the

Pixhawk Support Package for embedded coder, which offers the possibility to generate C code from designed Simulink models that are compatible with the PX1 board.

6.2.1. Pixhawk support package. The Pixhawk Support Package allows users to convert Simulink models to readable, compact, and fast C and C++ code compatible for deployment on the PX1 flight management unit using the Pixhawk Toolchain. Using this support package, several embedded PX1 sensors can be interfaced with the Simulink flight code allowing for real-time sensor data to be used in-flight. These sensors include the gyroscope, IMU, internal attitude estimation and GPS. An input block allows for transmitter commands to be used in the code, while an output block allows for generated output signals to be send to the 14 PWM ports of the PX1. Status updates can be communicated with the appropriate blocks for the LED, multi-tone audio speaker and data logger. Figure 6.12 shows the library with the provided blocks of the Pixhawk Support Package in MATLAB 2016b.

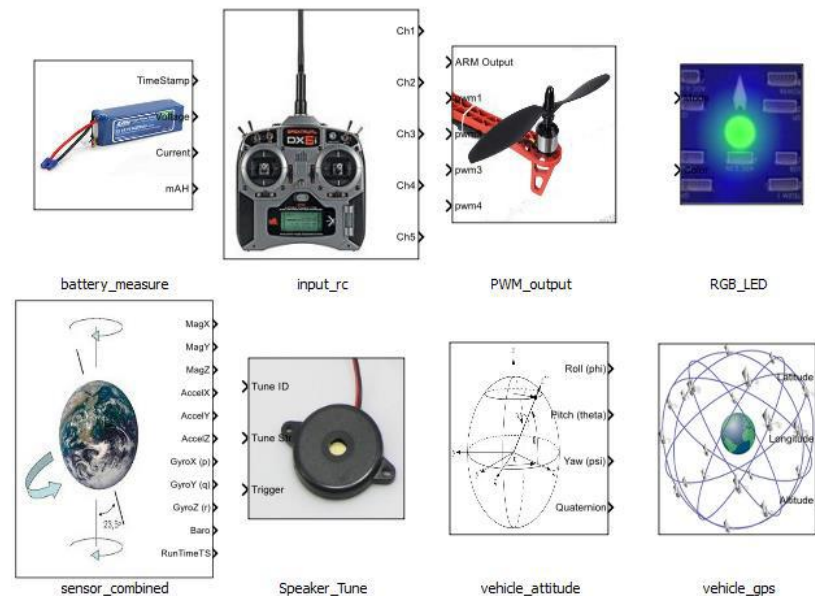


Figure 6.12. Pixhawk Support Package library blocks (Pilot Engineering Group Mathworks, 2017).

6.2.2. S-function development. The Pixhawk Support Package library (Figure 6.12) provides many blocks allowing for the communication between the Simulink code and the PX1 internal sensors and output ports. However, some blocks required for the application to the Flamewheel are not provided and were developed at the Advanced Dynamics and Control Laboratory (ADCL). One approach to design these blocks is using S-Function coder for Simulink. An S-Function is a representation of a Simulink block written in C or C++ code that can be executed to create a desired interface between Simulink and the PX1. Figure 6.13 shows developed S-Function blocks at the ADCL required for the implementation of the developed control algorithms in the Flamewheel quadrotor (Rivera K. , 2018). The ‘PX4_signal_log’-block allows for flight data to be stored on the microSD card of the PX1 for post-flight data processing, the ‘PX4_gps_position’ and ‘PX4_gyros_filtered’ blocks provide readings of the filtered and corrected readings for the GPS and gyroscope, and the ‘laser’-block allows for the readings from the Lightware SF11/C laser to be used in the Simulink code during flight.

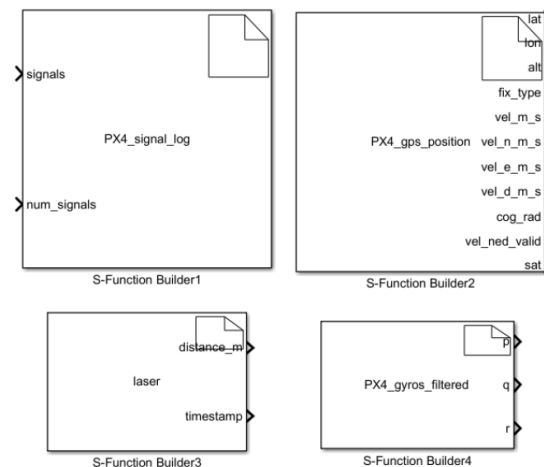


Figure 6.13. S-Functions developed at the ADCL (Rivera K. , 2018).

6.2.3. Flamewheel F330 flight code. Figure 6.14 shows the general architecture of the implementation code used in the Flamewheel quadrotor for flight testing. The flight code is a modification of the simulation code discussed in Section 5.1. The ‘px4-SENSORS’ and ‘Attitude Estimation’ blocks in Figure 6.14 use blocks from the Pixhawk Support Package library (Figure 6.12) and developed S-Functions (Figure 6.13) to allow the inflow of real-time sensor data from the IMU, gyroscope, magnetometer, GPS and laser sensor into the code, which is subsequently used to estimate current states of the Flamewheel quadrotor. Estimated states are used in the ‘Control Laws’ and ‘Control Allocation’ blocks to control the quadrotor in a stable manner during flight. Transmitter inputs enter the code in the ‘RC Inputs’ subsystem which contains the transmitter block from the Pixhawk Support Package. Outputs of the code are sent to the LED, multi-tone audio and output ports of the PX1 in the ‘Pixhawk Outputs’ block. Desired states are logged on the PX1 microSD card in the ‘DATA RECORDING’-block using the developed S-Function shown in Figure 6.12.

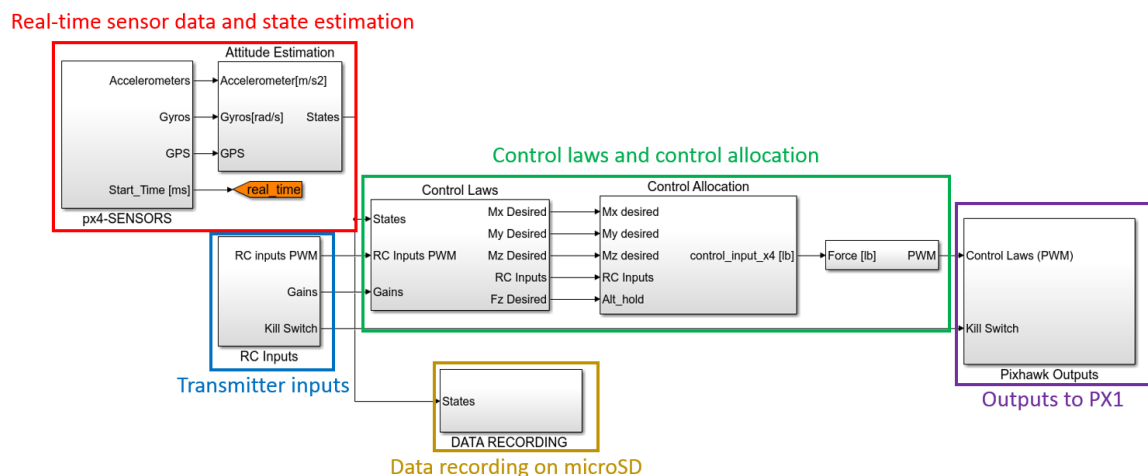


Figure 6.14. General architecture of the implementation code used in the Flamewheel quadrotor.

Note that the outer-loop tracking controller uses the cascade PID architecture presented in Section 4.1.2. The tuned virtual controller gains for the baseline NLDI in the flight code to achieve stable and desired transient and steady-state characteristics can be seen in Table 6.2.

Table 6.2

Baseline NLDI Flight Code Virtual Controller Tuned Gains

Inner Stability Fast Loop		
Controller	PID gains	Equation (4.2)
U_p, U_q	[8.8713 3 0.175]	
U_r	[17 0 0]	
Inner Stability Slow Loop		
Controller	PID gains	Equation (4.5)
U_θ, U_ϕ	[3 0 0]	
U_ψ	[1.4 0 0]	
Outer Tracking loop		
Controller	PID gains	Equation (4.11)
K_u, K_v	[6.62 1.55 0]	
K_w	[1.369 0.17 0.12]	
$K_{\dot{x}}, K_{\dot{y}}$	[0.5 0 0]	Equation (4.12)
$K_{\dot{z}}$	[0.22 0.06 0]	

6.3. Ground Weather Station

A characterization of the current wind conditions is required in outdoor flight testing in order to evaluate the Flamewheel performance with the developed control algorithms as a function of the current wind conditions. This can be accomplished by

using the ULTIMETER 2100 weather station developed by Peet Bros which offers a wide variety of weather measurement options such as humidity, amount of rain, temperature and wind, as can be seen in the wire diagram shown in Figure 6.15.

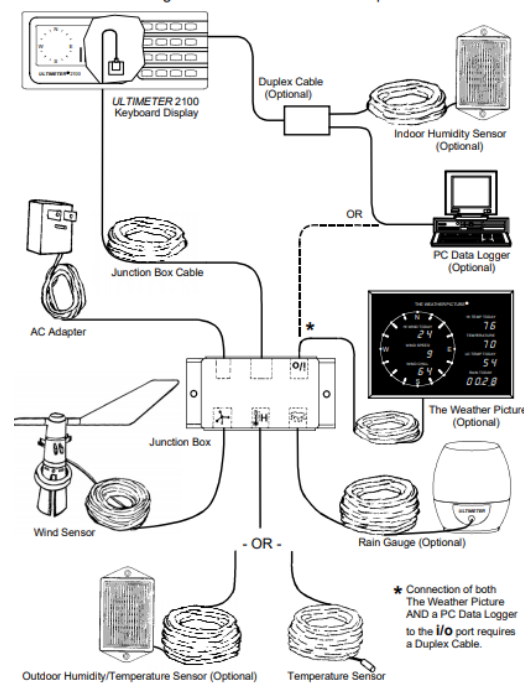


Figure 6.15. ULTIMETER 2100 wiring diagram showing all compatible sensors (Bros, 2017).

The ULTIMETER 2100 weather station components required in this research include the keyboard/display unit that provides multiple options to show current weather statistics and an anemometer with wind vane that can be connected to the keyboard/display unit with a 40ft cable as can be seen in (Figure 6.16).

Weather measurements from the ULTIMETER 2100 can be stored using the WeatherText software from Peet Bros. This software saves various measurements in one minute intervals in a .txt file on a connected computer. A sample report can be seen in Figure 6.17, which shows the collection of all current measurements, highest wind speed

over the past minute, highest wind speed since data collection started and one minute average wind speed.



Figure 6.16. ULTIMETER 2100 ground weather station.

```

ULTIMETER WEATHER REPORT 01/01/09 12:05A
Wind:  Cur   3.2MPH 328Deg, 1mAvg  4.0MPH,  1mPeak  6.8MPH 316Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  70.8F, Hi  71.7F, Lo  70.8F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

ULTIMETER WEATHER REPORT 01/01/09 12:06A
Wind:  Cur   0.9MPH 264Deg, 1mAvg  2.3MPH,  1mPeak  3.6MPH 309Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  70.6F, Hi  71.7F, Lo  70.6F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

ULTIMETER WEATHER REPORT 01/01/09 12:07A
Wind:  Cur   2.3MPH 329Deg, 1mAvg  3.3MPH,  1mPeak  4.7MPH 264Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  70.5F, Hi  71.7F, Lo  70.5F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

ULTIMETER WEATHER REPORT 01/01/09 12:08A
Wind:  Cur   3.8MPH 309Deg, 1mAvg  2.6MPH,  1mPeak  4.3MPH 322Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  70.3F, Hi  71.7F, Lo  70.3F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

ULTIMETER WEATHER REPORT 01/01/09 12:09A
Wind:  Cur   5.4MPH 314Deg, 1mAvg  5.2MPH,  1mPeak  7.3MPH 291Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  70.0F, Hi  71.7F, Lo  70.0F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

ULTIMETER WEATHER REPORT 01/01/09 12:10A
Wind:  Cur   3.3MPH 302Deg, 1mAvg  4.3MPH,  1mPeak  6.9MPH 304Deg
      Hi   7.3MPH 323Deg
Temp In: Cur  69.7F, Hi  71.7F, Lo  69.7F
Rain:  Today  0.00in, Since 01/01/09:  0.00in

```

Figure 6.17. Sample WeatherText software wind measurement report.

6.4. Flight Testing Locations

Flight tests with the Flamewheel UAV quadrotor were performed at two different locations. Initial flight testing was performed at the Indoor Unmanned Systems Testing Facility (IUSTF) at the John Mica Engineering & Aerospace Innovation Complex (MicaPlex) for initial tuning and nominal flight evaluation. Outdoor flight testing was performed at the artificial turf softball field at Embry-Riddle Aeronautical University to evaluate the Flamewheel flight under wind disturbance conditions.

6.4.1. MicaPlex indoor unmanned systems testing facility. The IUSTF is an indoor multipurpose testing environment located inside the Micaplex at the Embry-Riddle Aeronautical University Research Park (Figure 6.18). The IUSTF offers a multipurpose indoor testing environment for ground and air unmanned systems. The facility is equipped with a VICON motion capture system, which allows for the capturing and tracking of the motion of both ground and air systems inside the facility. The VICON motion capture system software can be interfaced with MATLAB/Simulink to allow for the use of captured data in both real-time and post processing applications.

Initial flight tests conducted inside the IUSTF were aimed towards tuning the inner stability loop virtual controllers of the baseline NLDI control law under nominal no wind disturbance conditions. The IUSTF offers a near disturbance free environment with only small disturbances caused by ground and wall effects. Subsequent flight tests evaluated the performance of the controllers under indoor nominal conditions in the Flamewheel quadrotor. The indoor flight test setup can be seen in Figure 6.19.



Figure 6.18. Location of Indoor Unmanned Systems Testing Facility inside the MicaPlex (Google, 2019).



Figure 6.19. Flight testing inside the Indoor Unmanned Systems Testing Facility.

6.4.2. ERAU artificial turf softball field. Outdoor flight testing was conducted at the artificial turf softball field at Embry-Riddle Aeronautical University shown in Figure 6.20. The artificial turf softball field offers a favorable testing location due to the available space and its close proximity to the ADCL. Additionally, the artificial turf field remains functional after nightly precipitation as compared to real grass and offers a relatively soft and smooth surface for quadrotor landings. Flight testing at the artificial softball field was performed to evaluate the developed control algorithms in an outdoor wind environment, which could be characterized using the ground weather station.



Figure 6.20. Artificial Turf Softball Field at Embry-Riddle Aeronautical University (Google, 2019).

7. Implementation Analysis

The simulation results presented in Chapter 5 showed the improvement in position hold capabilities of the quadrotor UAV simulation with the extended nonlinear dynamic inversion (NLDI), and the improvement in rotational stability gained by augmenting the baseline NLDI with adaptive artificial neural networks (ANN) and \mathcal{L}_1 output-feedback adaptive control. This chapter will show flight test results of the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 output-feedback adaptive control augmentation implemented in the Flamewheel testbed introduced in Chapter 6.

No implementation results are shown for the extended NLDI because significant disadvantages of the extended NLDI application came to light in the simulation analysis performed in Chapter 5; a decrease in stability as compared to the baseline NLDI, the cancellation of the wind forces can only be performed in a full autonomous environment and the additionally required wind estimation/measurement method which, if not performed very accurately, will decrease the robustness of the overall system.

The implemented control law architectures are according to the presented architectures in Chapter 4. The outer loop of the NLDI control law will consist of the cascade PID architecture as presented in Section 4.1.2. Initial flight testing with the baseline NLDI augmented with ANN and \mathcal{L}_1 showed that augmenting the yaw rate channel in the fast loop did not significantly affect nor improve the dynamic response of the system, as it is also observed in the simulation results shown in Chapter 5. Yaw rate augmentation by the ANN and \mathcal{L}_1 in the fast loop is therefore omitted.

Since the adaptive ANN and \mathcal{L}_1 augmentation are designed and tuned in the

simulation environment discussed in Section 5.1, additional tuning is required for successful implementation in the flight code. This is achieved for the adaptive ANN by tuning the learning rates inside the adaptive ANN with a knob on the transmitter in flight before engaging the autonomous hold. The \mathcal{L}_1 adaptive controller is tuned by multiplying the augmentation value with an adjustable gain which will correct for the difference in magnitude of the virtual controllers in the simulation (Table 5.2) and implementation (Table 6.2) codes. This is required to effectively scale the output value of the \mathcal{L}_1 adaptive controller to achieve successful augmentation of the fast loop.

As discussed in Section 6.1.1, the PX1 runs a soft real-time operating system meaning that the internal sample time will depend on the required computation. In this application, the PX1 runs at a maximum sampling rate of 25Hz with all the flight code running effectively and consistently, but has been found to run as low as 13Hz on occasion. One point of concern related to this sample frequency is the design of the discrete \mathcal{L}_1 adaptive controller. Section 4.4.3 showed that the reference model is designed with $\omega_{n_m} = 25$ rad/s in order to mimic the ideal response of the fast loop of the baseline NLDI. According to the Nyquist frequency theorem, a PX1 board with a sampling rate of 25Hz will be able to correctly represent discretized dynamic signals with a maximum frequency of 12.5Hz (Ogata, 1995). Signals with a higher frequency than 12.5Hz will not be represented correctly in the PX1 board with a sampling rate of 25Hz. For a PX1 sampling rate of 13Hz, the highest supported frequency drops to 6.5Hz. The discretized reference model of the \mathcal{L}_1 adaptive controller has a natural frequency of $\omega_{n_m} = 25$ rad/s $= 4$ Hz. The desired dynamics in the implemented reference model of the \mathcal{L}_1 adaptive controller should therefore be represented correctly in the PX1. This is not an issue in the

simulation environment since the sampling frequency was set at $T_s = 500\text{Hz}$. The PX1 board therefore enforces a considerable restriction on the implementation of the \mathcal{L}_1 adaptive controller.

In regards to the implemented filter in the \mathcal{L}_1 adaptive controller, Section 4.4.3 showed that the bandwidth is established at 105 rad/s which is equal to a 16.7Hz frequency. The PX1 board has a maximum sample frequency equal to 25Hz, so effectively the implemented \mathcal{L}_1 filter will filter high frequency signals caused by the wind disturbance while not affecting the dynamics of the desired reference model.

Three sets of flight tests are performed for which the results are presented to evaluate the flight performance of the implemented baseline NLDI, baseline NLDI augmented with ANN and baseline NLDI augmented with \mathcal{L}_1 in the Flamewheel quadrotor. The first set of flight tests shows results for the three controllers inside the Indoor Unmanned Systems Testing Facility (IUSTF) to evaluate the performance of the controllers in a near disturbance free environment. The second set of flight tests show results for the three controllers implemented in the Flamewheel quadrotor in flight tests performed at the artificial turf softball field at minimal wind conditions. Finally, flight test results are shown for the Flamewheel quadrotor with the three controllers under low, medium and high wind disturbance conditions characterized by the ground weather station.

7.1. Performance Evaluation Indoor Nominal Conditions

The first set of flight tests focus on evaluating the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation in the IUSTF. This will assess the performance of the three controllers under nominal, almost disturbance

free indoor conditions. Ideally, the augmentation of the baseline NLDI with ANN and \mathcal{L}_1 should add robustness in the presence of uncertainties and disturbances but should not hurt the nominal performance of the baseline NLDI. The results for the nominal indoor tests for the three controllers should therefore be consistent, indicating that the addition of the augmentation does not negatively influence the performance of the baseline NLDI at nominal conditions.

The indoor testing consisted of an autonomous altitude hold and no position hold since the IUSTF does not allow for a GPS signal to be received by the GPS module. In this case, the pilot takes-off and engages a switch for the Flamewheel to sample-and-hold the current altitude autonomously while the pilot attempts to maintain the position manually. The performance metric for trajectory tracking described in Equation (5.4) was modified to reflect this limitation of the indoor environment to only include vertical displacement tracking error with $w_2 = w_3 = 0.4$ and $w_1 = w_4 = 0.1$ as defined in Equation (5.8).

Table 7.1 shows the performance metric results for comparable nominal indoor flight tests with the three controllers. The learning rates of the adaptive ANN introduced in Equation (4.54) are tuned for high adaption with $\Gamma_1 = \Gamma_2 = 33.5$ for the shown test results. The augmented output of the \mathcal{L}_1 adaptive controller was multiplied by 5.2 to obtain the desired response, which is also roughly the scaling difference between the virtual controllers in the fast loop in the simulation environment and in the flight code (Table 5.2 and Table 6.2).

As it can be seen, the baseline NLDI results in the best overall performance. The application of NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation both

result in a slightly worse performance as compared to the baseline NLDI. This is not alarming whatsoever though, and it is a commonly seen trait of the addition of adaptive control at nominal conditions. It is important to note that the baseline NLDI does not become unstable in nominal conditions with the application of adaptive ANN and \mathcal{L}_1 adaptive control. The effect of the adaptive controllers will primarily be seen during abnormal and disturbed conditions.

Table 7.1

Performance Index Comparison for Nominal Indoor Flight Tests

Controller	PM Attitude	PM Rates	PM Altitude	PM PWM	Global PI	Flight Time (s)
Baseline NLDI	0.8810	0.8038	0.8836	0.728300	0.8351	36.85
NLDI + NN	0.7325	0.7652	0.8719	0.726816	0.7589	33.50
NLDI + L1	0.8622	0.7224	0.8699	0.727104	0.7935	35.62

7.1.1. Baseline NLDI. In order to further analyze the performance of the adaptive ANN and \mathcal{L}_1 augmentation in the fast loop of the NLDI, first the baseline NLDI characteristics need to be known. Figure 7.1 shows the baseline NLDI roll and pitch rate virtual control values defined in Equation (4.2) for the indoor test flights during the autonomous hold. As it can be seen, both signals show a fast changing response close to zero to obtain the desired closed-loop response in the inner stability loop inversion using the baseline NLDI control law.

7.1.2. Adaptive ANN augmentation. Figure 7.2 shows the outputs of the adaptive ANN in the roll and pitch channel as defined in Equation (4.53). Figure 7.2 is extended to include the portion of the flight before the autonomous hold to show the behavior of the adaptive ANN after engagement. This behavior of the ANN after

activation can be observed in all flight tests with the Flamewheel, and so this portion will be omitted in future plots of flight test data to strictly evaluate the performance of the Flamewheel during autonomous altitude (and position) hold. Figure 7.2 shows that the ANN outputs attain an equilibrium with the baseline NLDI controller shortly after being activated and remain constant throughout the autonomous hold. The ANN outputs attain a constant value that is directly dependent on the selection of the e-modification parameter that counteracts uncontrolled parameter growth introduced in Equation (4.54). This parameter is set to 0.02 in all test flights shown in this chapter. The addition of this term will effectively enforce an upper bound on the outputs of the ANN. The ANN outputs do not show high frequency corrections due to the near disturbance free environment in the IUSTF.

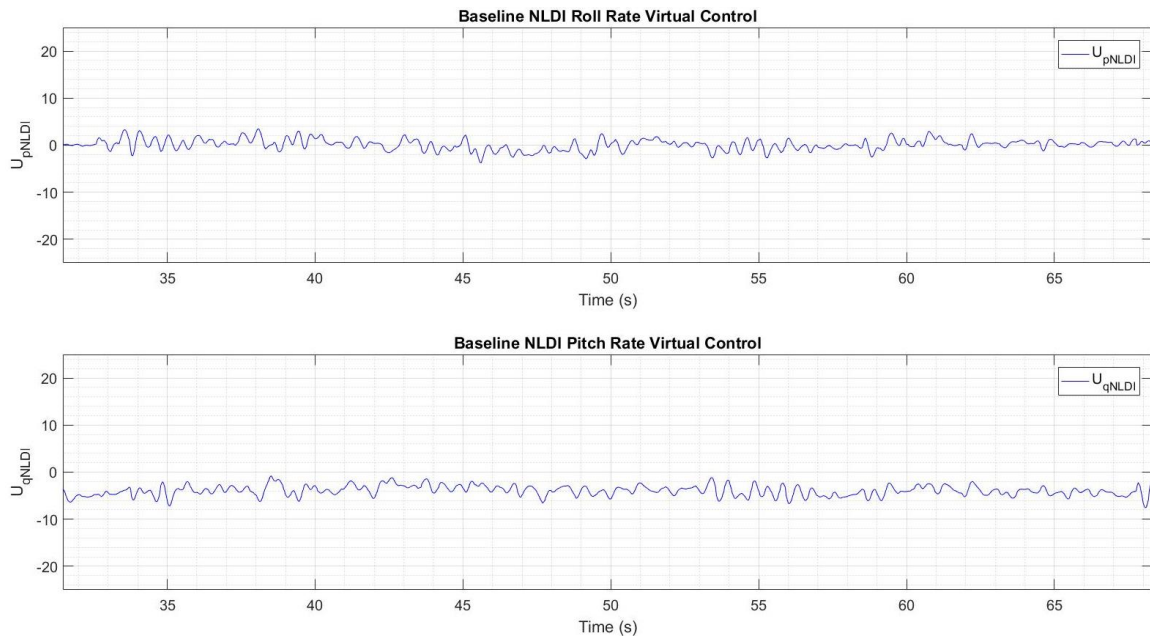


Figure 7.1. Baseline NLDI roll and pitch rate virtual control for nominal indoor flight testing.

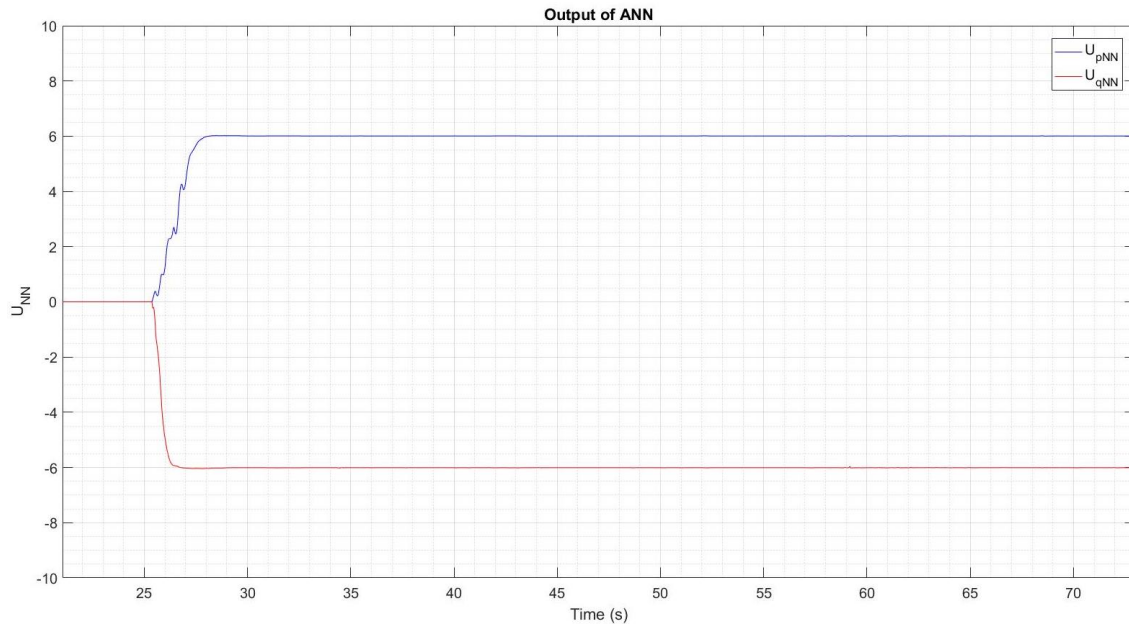


Figure 7.2. Outputs of the adaptive ANN for nominal indoor flight testing.

Figure 7.3 shows the virtual control signal generated by the NLDI with ANN augmentation as defined in Equation (4.58). Comparing Figure 7.3 with Figure 7.1 shows that the combined signal for the $U_{NLDI} + U_{NN}$ in Figure 7.3 is very similar to the U_{NLDI} signal in the baseline NLDI shown in Figure 7.1. This shows that the addition of adaptive ANN augmentation does not degrade the nominal behavior of the NLDI control law.

Comparing Figure 7.3 and Figure 5.19 shows a difference between the NLDI with adaptive ANN augmentation virtual control in simulation and implementation. This is caused by the selection of the e-modification parameter (Equation (4.54)), which is set at the same value for the simulation and implementation analyses. Because of the upper bound that is required to promote stability, the adaptive ANN is not able to drive the U_{pNLDI} and U_{qNLDI} signals to zero in the implementation results as is achieved in the simulation results. This creates an equilibrium between the various signals that still attains the desired dynamic characteristics as can be seen in Figure 7.3.

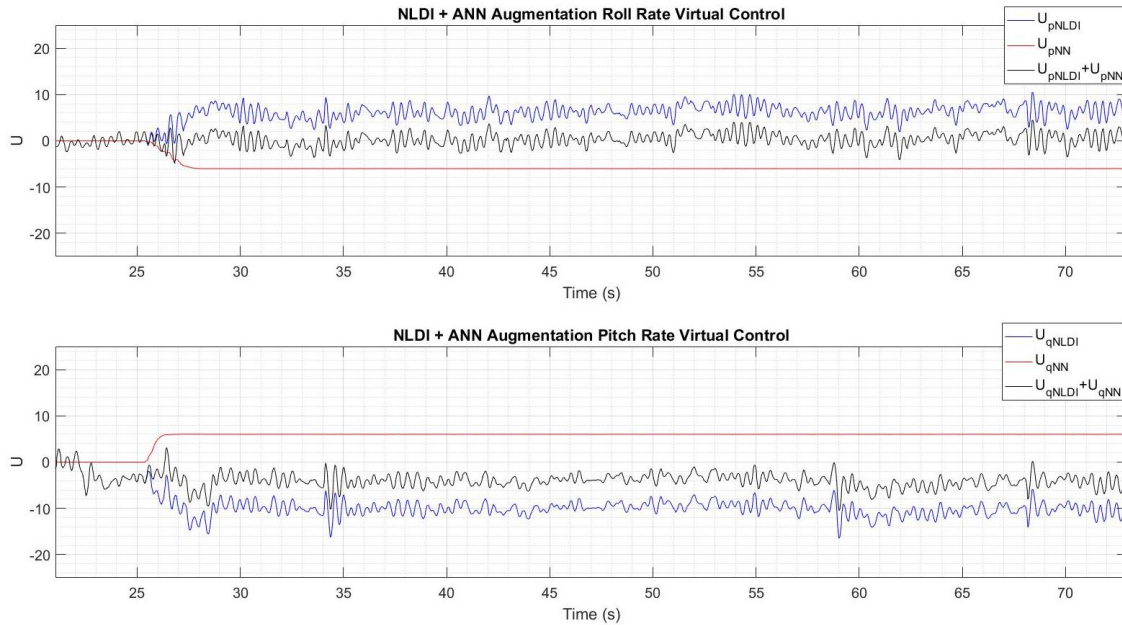


Figure 7.3. NLDI with adaptive ANN augmentation roll and pitch rate virtual control for nominal indoor flight testing.

As mentioned earlier, adaptive ANN are notorious for their sometimes unexpected behavior. This unwanted trait caused by the inherent self-adaptive nature of the network parameters can be seen in one of the performed test flights in the IUSTF. Figure 7.4 shows the outputs of the adaptive ANN for one of the test flights. Initially the ANN were stable and output constant values throughout the flight. However, at some point towards the end of the flight the adaptive ANN outputs show a high frequency character without an apparent reason. The result of this instability can be seen in Figure 7.5 which shows the NLDI with adaptive ANN augmentation roll and pitch rate virtual control signals becoming very oscillatory at the end of the autonomous altitude hold. In this flight, the pilot had to recover to prevent a loss of complete flight control. This shows the care that must be taken when introducing ANN in active flight controls.

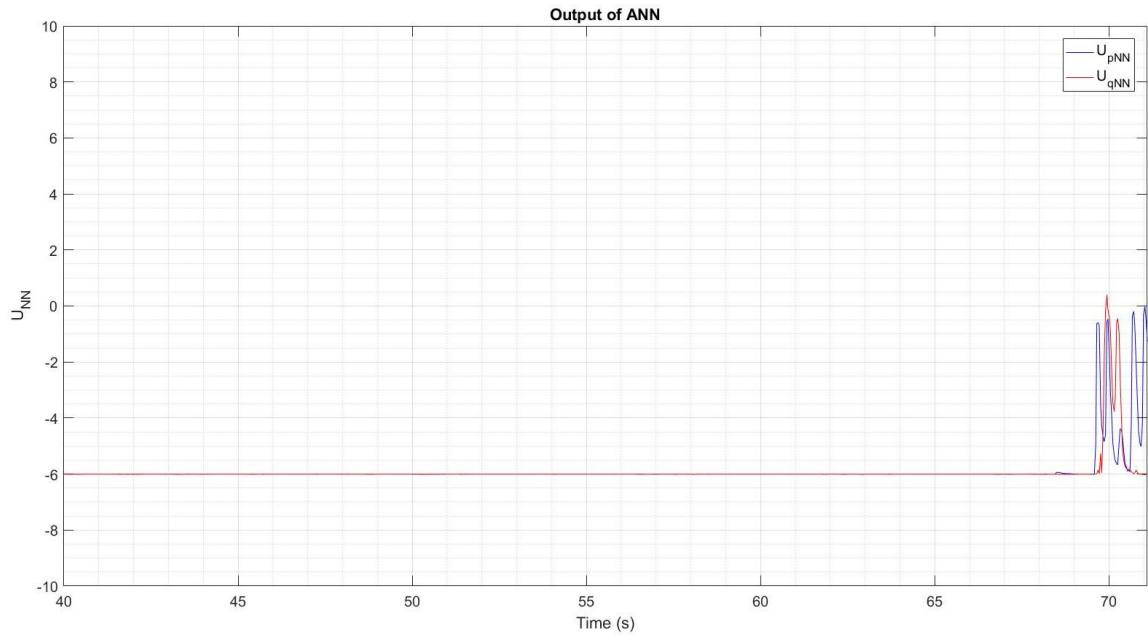


Figure 7.4. Outputs of the adaptive ANN showing instability for nominal indoor flight testing.

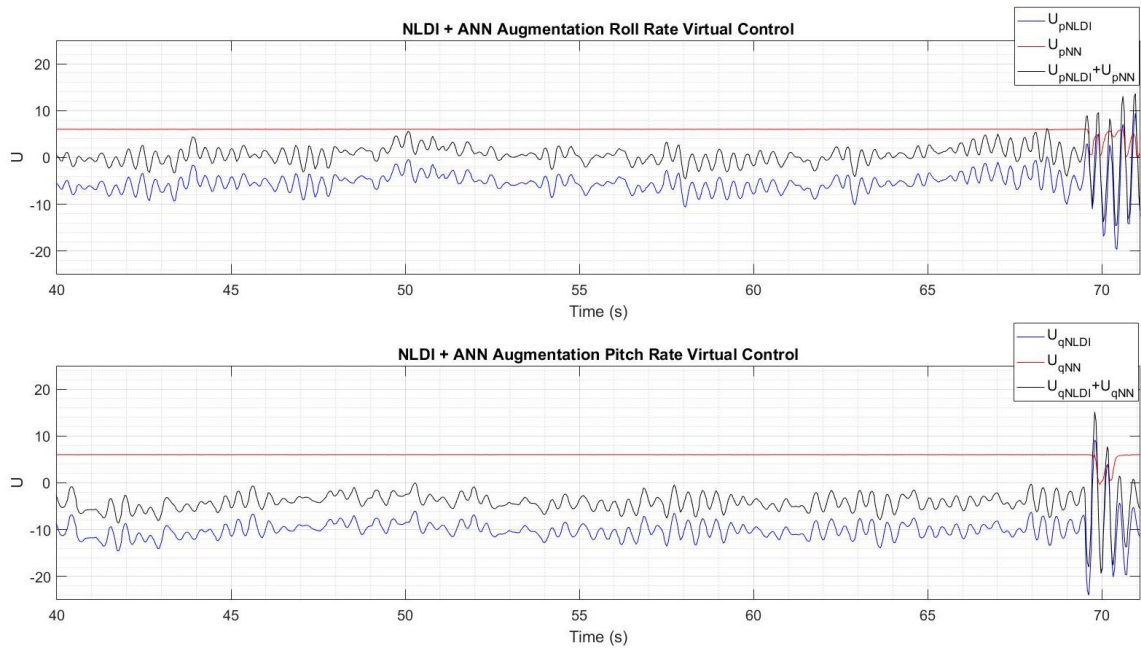


Figure 7.5. NLDI with adaptive ANN augmentation roll and pitch rate virtual control showing instability for nominal indoor flight testing.

7.1.3. \mathcal{L}_1 output-feedback adaptive control augmentation. Figure 7.6

shows the outputs of the \mathcal{L}_1 output-feedback adaptive controller in the roll and pitch channel as defined in Equation (4.101). Figure 7.7 shows the NLDI with \mathcal{L}_1 adaptive controller augmentation for the roll and pitch rate virtual controllers as defined in Equation (4.89). The results show that the \mathcal{L}_1 adaptive controller only augments small values, which is expected due to the near disturbance free environment in the IUSTF, resulting in small uncertainties in the feedback linearization in the fast loop of the NLDI control law.

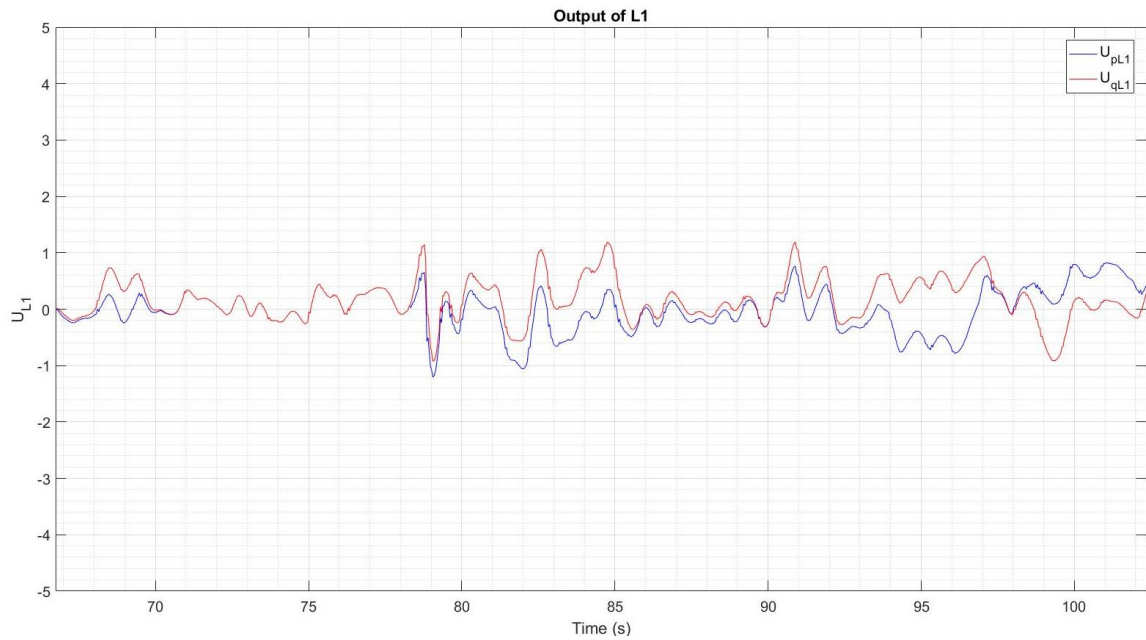


Figure 7.6. Outputs of the \mathcal{L}_1 adaptive controller for nominal indoor flight testing.

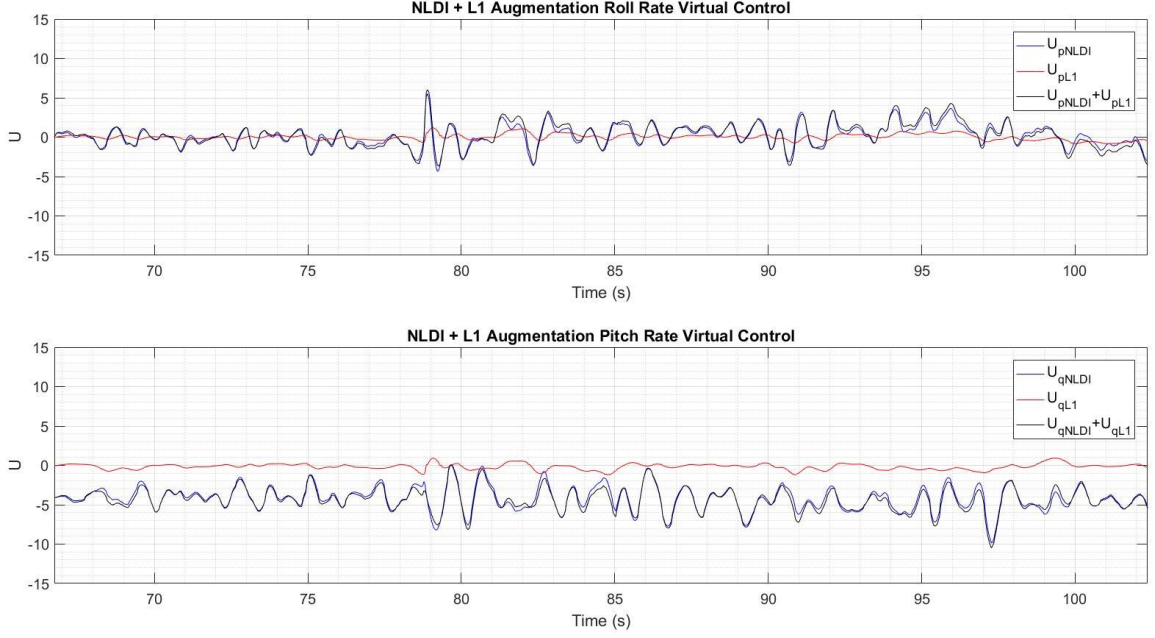


Figure 7.7. NLDI with \mathcal{L}_1 adaptive controller augmentation roll and pitch rate virtual control for nominal indoor flight testing.

7.2. Outdoor Performance Evaluation

The second set of flight tests focus on evaluating the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation in the Flamewheel quadrotor at the artificial turf softball field at different magnitude wind conditions characterized by the ground weather station. The outdoor testing campaign consists of an autonomous altitude and position hold. During these tests, the pilot takes-off and engages a switch for the Flamewheel to sample-and-hold the current altitude and position autonomously. The original performance metric for trajectory tracking defined in Equation (5.4) is applied with $w_1 = w_2 = w_3 = 0.3$ and $w_4 = 0.1$ as defined in Equation (5.8).

7.2.1. Nominal wind disturbance conditions. While the results in Section 7.1 show the nominal results in the indoor near disturbance free environment, this section

shows nominal results in the outdoor environment. The flight tests presented here are performed at near wind free conditions as measured by the ground weather station to allow for a performance evaluation at outdoor nominal conditions.

Table 7.2 shows the performance metric results for comparable nominal outdoor flight tests with the three controllers. The learning rates of the adaptive ANN defined in Equation (4.54) are tuned to allow for fast adaptation with $\Gamma_1 = \Gamma_2 = 38.1$ for the shown test results. The augmented output of the \mathcal{L}_1 adaptive controller was multiplied by 8.85 to obtain the desired response.

Table 7.2

Performance Index Comparison for Nominal Outdoor Flight Tests

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI	Flight Time (s)	Average Wind Speed (kts)	Maximum Wind Peak (kts)
Baseline NLDI	0.9049	0.8530	0.7381	0.727220	0.8215	31.19	0.0	0.0
NLDI + NN	0.8591	0.7363	0.7269	0.729182	0.7696	27.17	0.0	0.0
NLDI + L1	0.9086	0.8164	0.7529	0.729835	0.8164	30.16	0.0	0.0

Table 7.2 shows similar results for the nominal outside tests as for the nominal inside test presented in Section 7.1. All three controllers result in a desired stable and tracking quadrotor dynamic response with the baseline NLDI slightly outperforming the NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation. Again, this is expected with the most important result in these tests being that the augmented controllers perform about as well as the nominal baseline controller at nominal outdoor flight conditions.

The baseline NLDI roll and pitch rate virtual control signal characteristics for the outdoor testing during the autonomous altitude and position hold are observed to be almost identical to the results obtained in the indoor testing shown in Figure 7.1. The

same can be said about the NLDI with ANN augmentation and NLDI with \mathcal{L}_1 augmentation which show to result in very similar looking response plots for the roll and pitch rate virtual controllers as shown in Figure 7.2, Figure 7.3, Figure 7.6 and Figure 7.7. This confirms the proper working of the augmented controllers at nominal outdoor conditions.

Figure 7.8 shows the 3D trajectory of the Flamewheel quadcopter with the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation during the autonomous altitude and position hold in the outdoor nominal environment. The colored spheres represent the commanded position and altitude for the three separate flights. As can be seen by the axis limits, all controllers hold the commanded position and altitude well with some slight transient error when switching from pilot-in-the-loop to autonomous control.

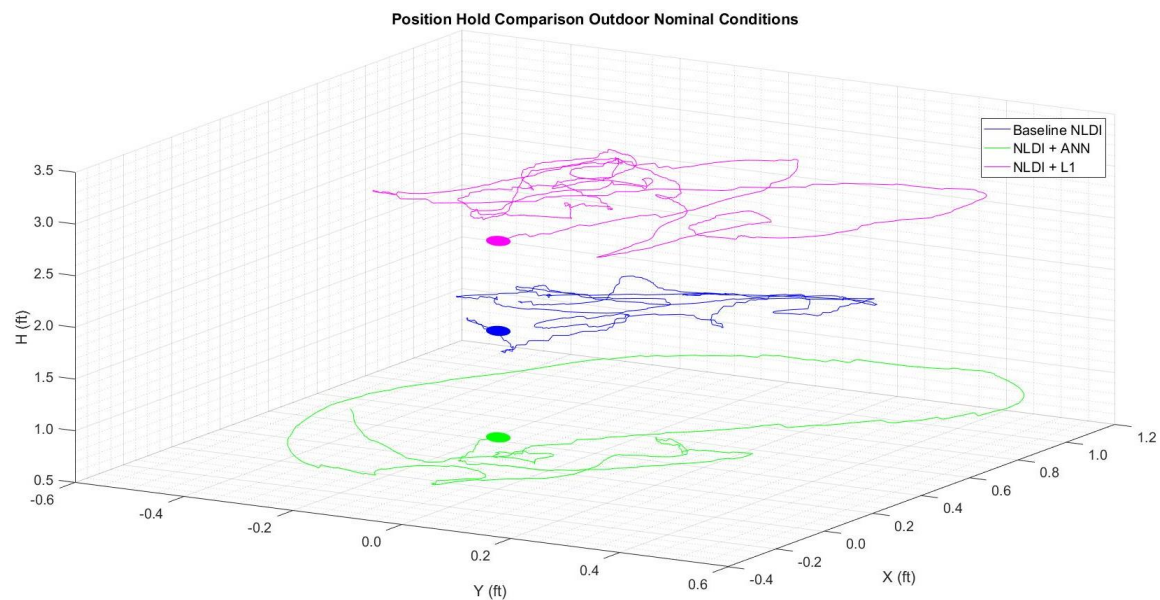


Figure 7.8. 3D trajectory comparison for nominal outdoor flight testing.

7.2.2. Low wind disturbance conditions. Table 7.3 shows the performance index comparison for low wind conditions for the three controllers together with the average and maximum wind speed peak for each flight. Table 7.3 indicates that for low wind conditions, which can be characterized by wind speeds up to 3kts, the three controllers perform uniformly. Comparing Table 7.3 with Table 7.2 shows that the presence of low wind disturbance results in a reduction of the global PI and all PMs for the three controllers.

Table 7.3

Performance Index Comparison for Low Wind Conditions

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI	Flight Time (s)	Average Wind Speed (kts)	Maximum Wind Peak (kts)
Baseline NLDI	0.8338	0.6120	0.6778	0.725926	0.7097	29.96	2.1	3.6
NLDI + NN	0.8346	0.6695	0.6014	0.730011	0.7047	29.32	1.8	2.3
NLDI + L1	0.8940	0.6410	0.6166	0.729872	0.7185	35.50	3.0	5.3

Figure 7.9 shows the 3D trajectory of the Flamewheel quadcopter with the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation during the autonomous altitude and position hold outdoors in the presence of low wind disturbance conditions. As it can be seen, the Flamewheel is able to sustain controlled flight with all three controllers. Comparing Figure 7.9 with Figure 7.8 shows that the overall deviation of the quadrotor from the desired waypoint is larger for higher wind disturbance conditions.

Adaptive ANN augmentation. Figure 7.10 shows the outputs of the adaptive ANN in the roll and pitch channel for low wind conditions during the autonomous position and altitude hold. The learning rates of the adaptive ANN introduced in Equation

(4.54) are tuned for high adaption with $\Gamma_1 = \Gamma_2 = 30.0$ for the shown test results. A similar constant offset of the outputs as seen in Section 7.1.2 can be observed.

Superimposed higher frequency spikes can be seen which are induced by the ANN in an effort to correct for the wind disturbances in the fast loop of the NLDI control law.

\mathcal{L}_1 output-feedback adaptive control augmentation. Figure 7.11 shows the outputs of the \mathcal{L}_1 output-feedback adaptive controller in the roll and pitch channel for low wind conditions during the autonomous position and altitude hold. The augmented output of the \mathcal{L}_1 adaptive controller was multiplied by 3.1 to obtain the desired response. The outputs of the \mathcal{L}_1 output-feedback adaptive controller show small, high frequency values which are augmented in the fast loop in an effort to increase the robustness of the rotational dynamics response.

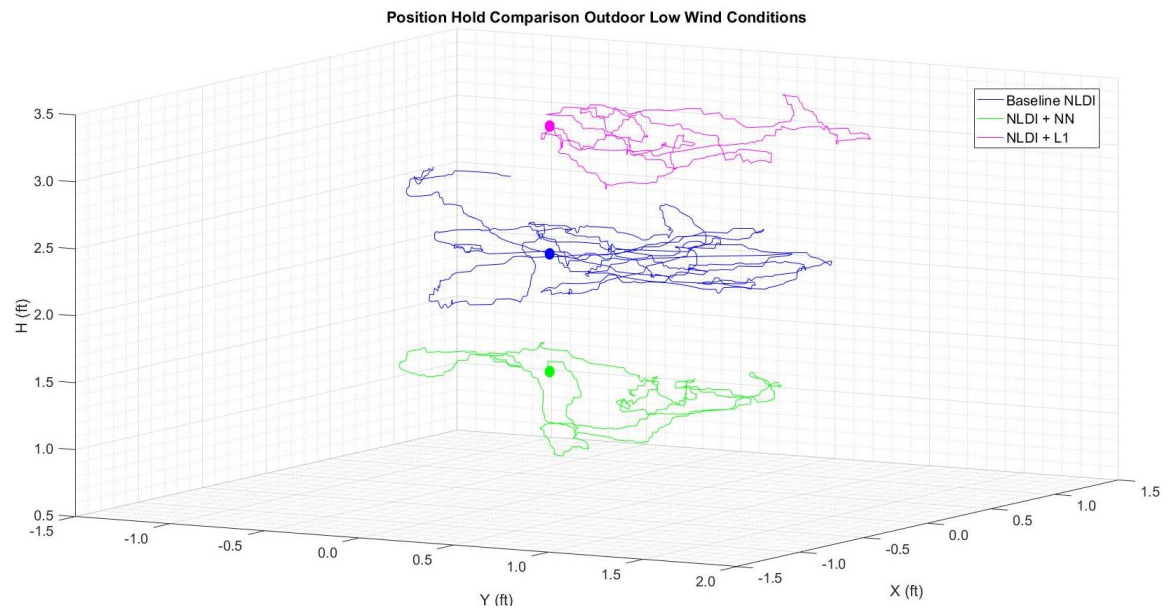


Figure 7.9. 3D trajectory comparison for low wind conditions.

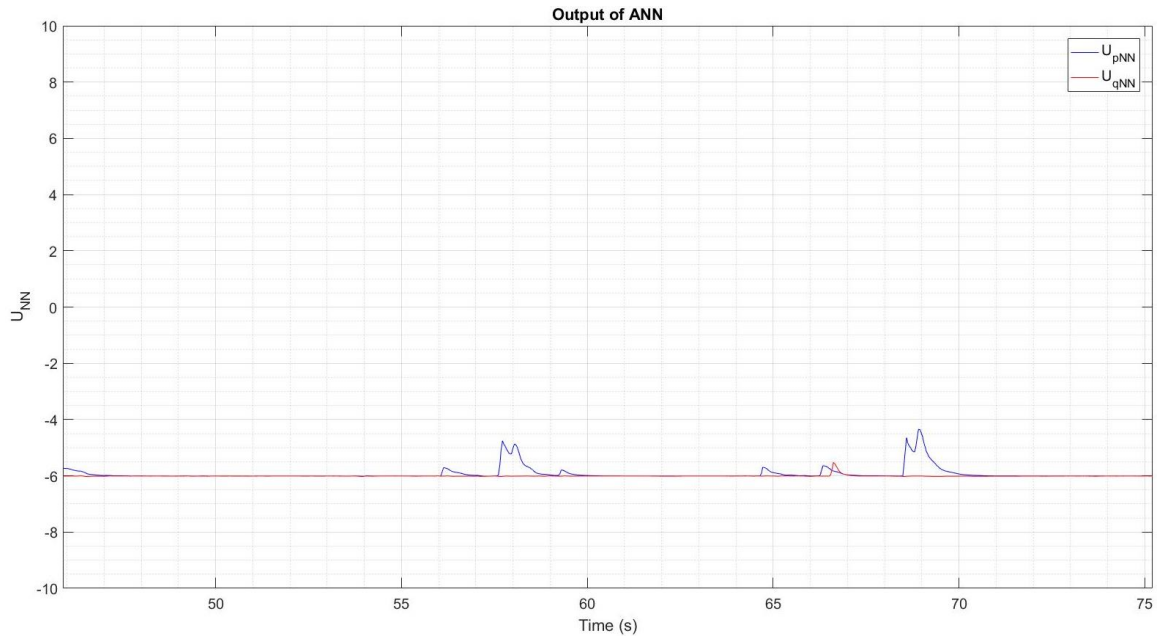


Figure 7.10. Outputs of the adaptive ANN for low wind conditions.

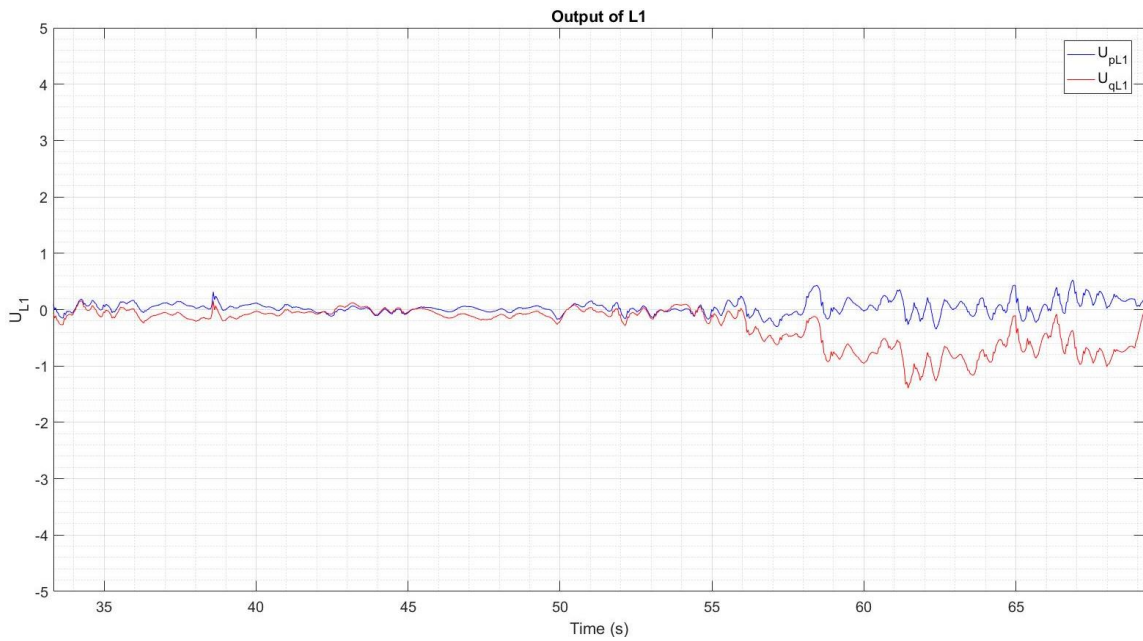


Figure 7.11. Outputs of the \mathcal{L}_1 adaptive controller for low wind conditions.

7.2.3. Medium wind disturbance conditions. Table 7.4 shows the performance index comparison for medium wind conditions for the three controllers

together with the average and maximum wind speed peak for each flight. Table 7.4 indicates that for medium wind conditions, which can be characterized by wind speeds between 3kts and 5kts, the baseline NLDI with ANN and \mathcal{L}_1 augmentation outperform the baseline NLDI controller. The added value of augmenting the baseline NLDI control law can clearly be seen for flights exposed to increased wind disturbance conditions.

The performance index results at medium wind conditions show that the previous discussed concern in regards to the lower than desired sampling frequency of the reference model of the \mathcal{L}_1 output-feedback adaptive controller does not degrade the robustness of the system.

Table 7.4

Performance Index Comparison for Medium Wind Conditions

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI	Flight Time (s)	Average Wind Speed (kts)	Maximum Wind Peak (kts)
Baseline NLDI	0.6735	0.0958	0.4333	0.729901	0.4338	19.04	4.0	6.4
NLDI + NN	0.7918	0.4697	0.5243	0.727829	0.6085	27.45	3.6	5.0
NLDI + \mathcal{L}_1	0.7964	0.4569	0.5586	0.729209	0.6165	16.16	3.6	4.6

Figure 7.12 shows the 3D trajectory of the Flamewheel quadrotor with the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation during the autonomous outdoor altitude and position hold in the presence of medium wind disturbance conditions. In this case, the Flamewheel is able to sustain controlled flight with all three controllers. Comparing Figure 7.12 with Figure 7.9 shows that the overall deviation of the quadrotor from the desired waypoint is larger for increased wind disturbance conditions.

A closer look can be taken at the stability of the angular dynamics performance of

the three controllers in the presence of medium wind disturbance conditions.

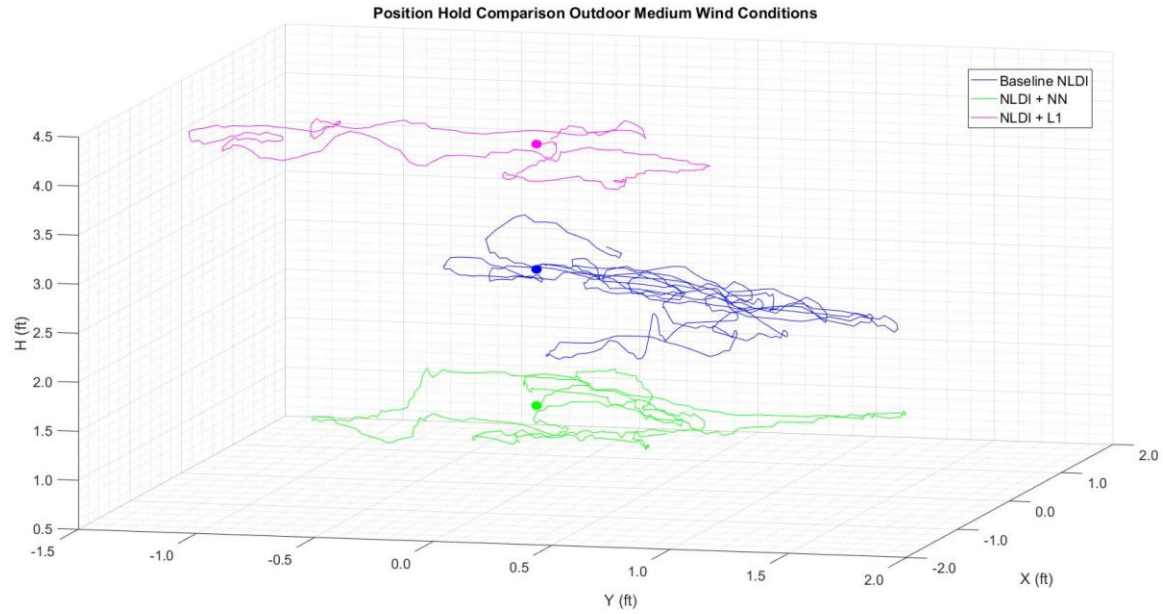


Figure 7.12. 3D trajectory comparison for medium wind conditions.

Baseline NLDI. Figure 7.13 shows the roll and pitch angle response for the baseline NLDI control law under medium wind disturbance conditions. Both responses show a very oscillatory behavior of the angular dynamics of the Flamewheel quadrotor during the 3D autonomous position and altitude hold. The quadrotor with baseline NLDI is heavily affected by the wind disturbance and the robustness of the rotational dynamics is decreased.

Adaptive ANN augmentation. Figure 7.14 shows the outputs of the adaptive ANN during the position and altitude hold at medium wind conditions. The learning rates of the adaptive ANN are tuned for high adaption with $\Gamma_1 = \Gamma_2 = 46.1$ to allow for fast adaptation of the network parameters to the wind disturbance. As can be seen, a constant offset of the outputs is again observed. Comparing Figure 7.14 with Figure 7.10 shows the increase of high frequency characteristics of the outputs of the adaptive ANN at

medium wind conditions as compared to low wind conditions. For an increase in external disturbance induced by the wind, the adaptive ANN will attempt to estimate and adapt for the external uncertainties in the system faster.

Figure 7.15 shows the effect of the adaptive ANN augmentation on the roll and pitch angle response under medium wind conditions during the autonomous position and altitude hold. Comparing Figure 7.15 with Figure 7.13 shows the increase in robustness of the rotational dynamics of the Flamewheel quadrotor in flight exposed to medium wind conditions. The relatively large oscillatory behavior observed in the Flamewheel with the baseline NLDI control law have been considerably corrected for by the addition of ANN augmentation.

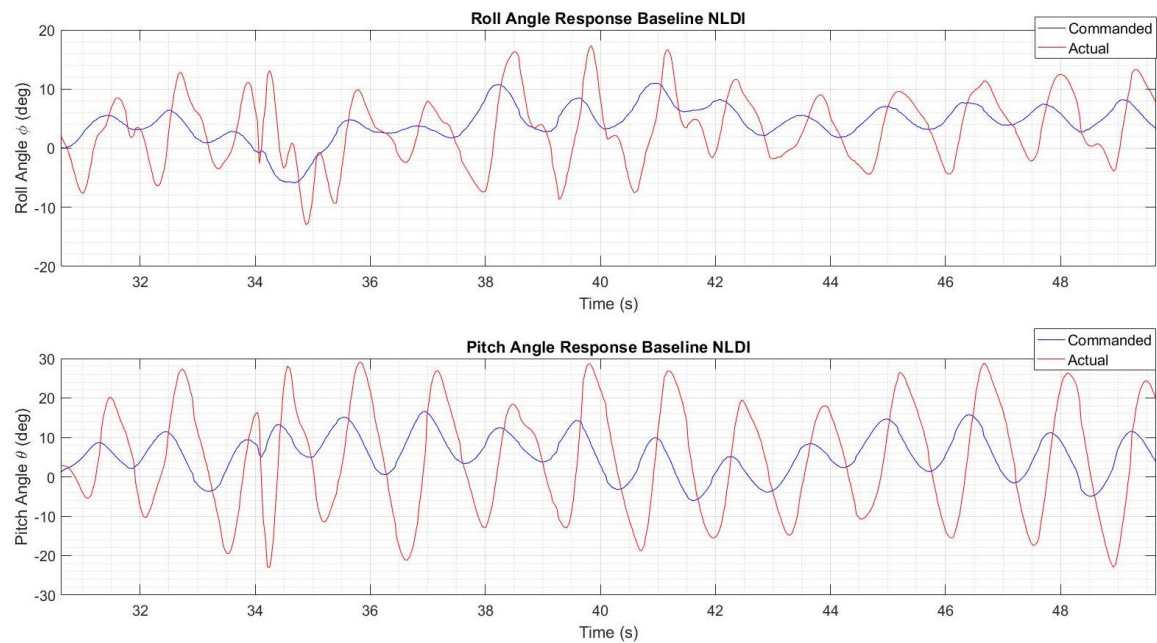


Figure 7.13. Roll and pitch angle response for baseline NLDI under medium wind conditions.

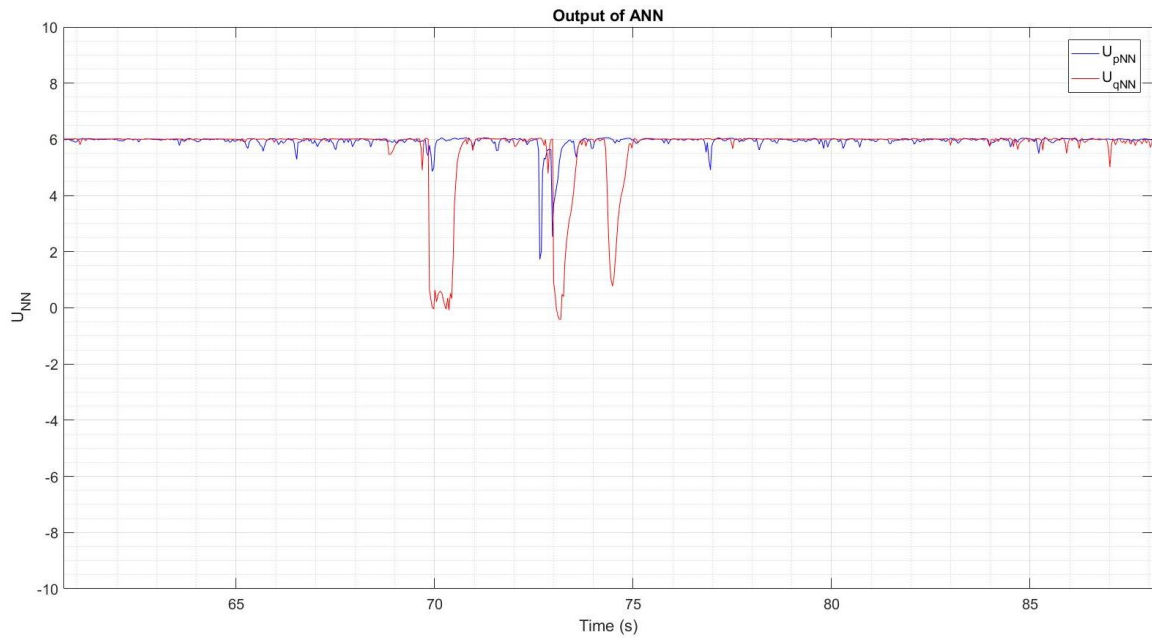


Figure 7.14. Outputs of the adaptive ANN for medium wind conditions.

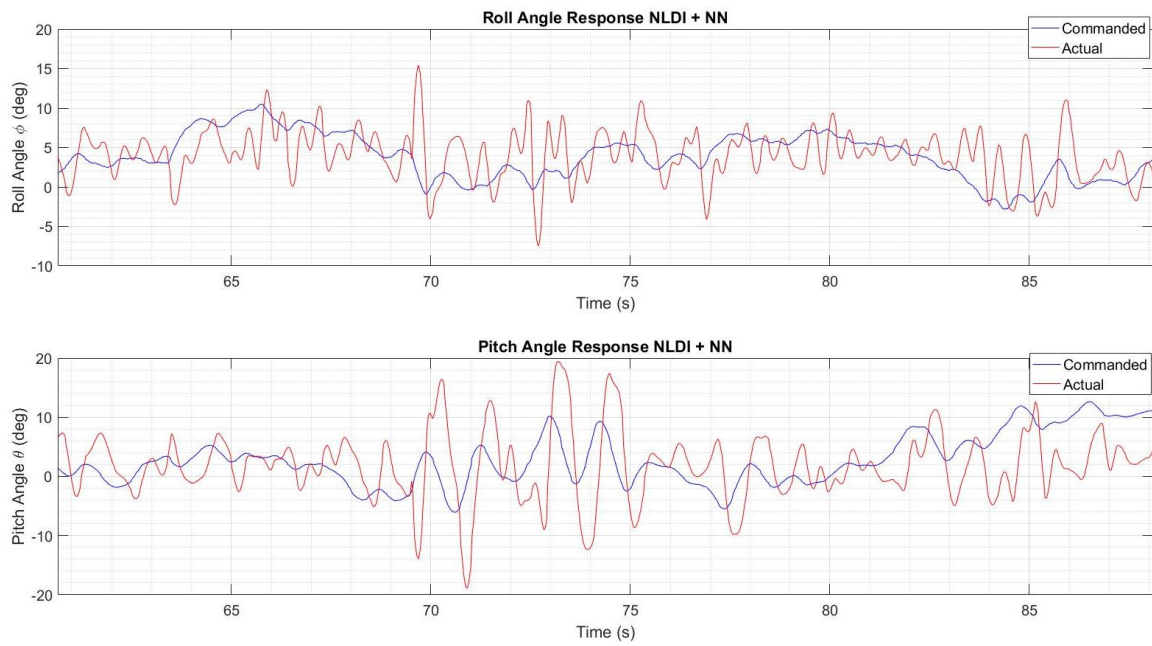


Figure 7.15. Roll and pitch angle response for baseline NLDI with adaptive ANN augmentation under medium wind conditions.

\mathcal{L}_1 output-feedback adaptive control augmentation. Figure 7.16 shows the outputs of the \mathcal{L}_1 output-feedback adaptive controller in the roll and pitch channel for medium wind conditions during the autonomous position and altitude hold. The augmented output of the \mathcal{L}_1 adaptive controller was multiplied by 2.1 to obtain the desired response. Comparing Figure 7.16 with Figure 7.11 shows a similar output at medium wind conditions as at low wind conditions. The \mathcal{L}_1 adaptive controller is attempting to aid the overall rotational dynamics robustness of the quadrotor by augmenting values to correct for the wind disturbance.

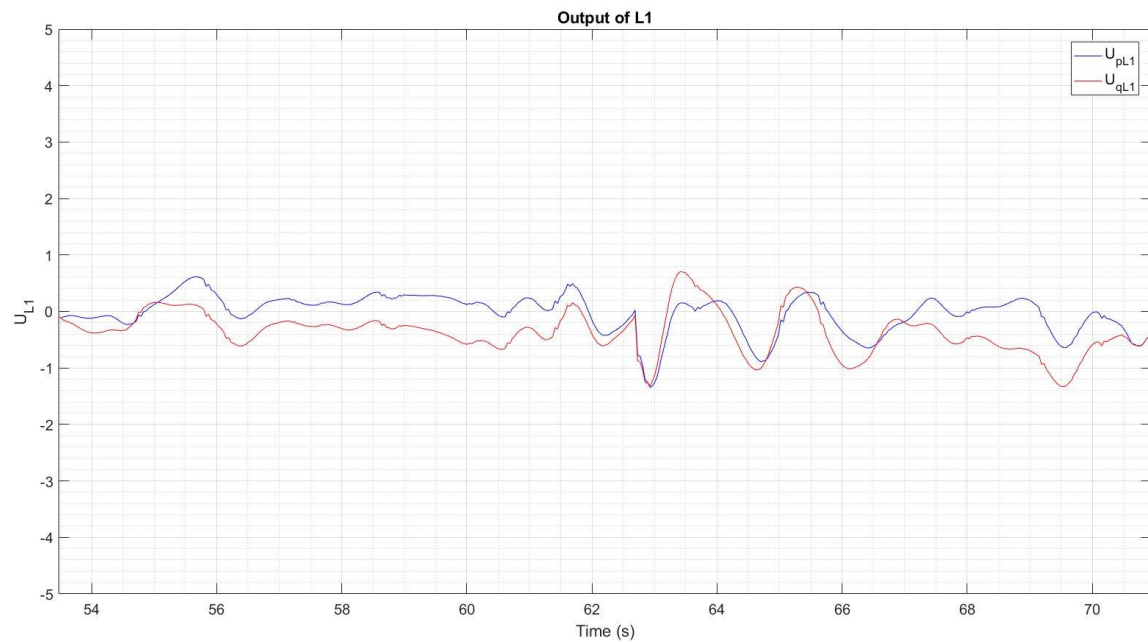


Figure 7.16. Outputs of the \mathcal{L}_1 adaptive controller for medium wind conditions.

Figure 7.17 shows the effect of the \mathcal{L}_1 adaptive control augmentation on the roll and pitch angle response under medium wind conditions during the autonomous position and altitude hold. Comparing Figure 7.17 with Figure 7.13 shows the increase in robustness of the rotational dynamics of the Flamewheel quadrotor in flight exposed to

medium wind conditions. Although an oscillatory behavior can still be seen, the magnitude and frequency of the oscillations have been decreased by the addition of \mathcal{L}_1 adaptive control augmentation.

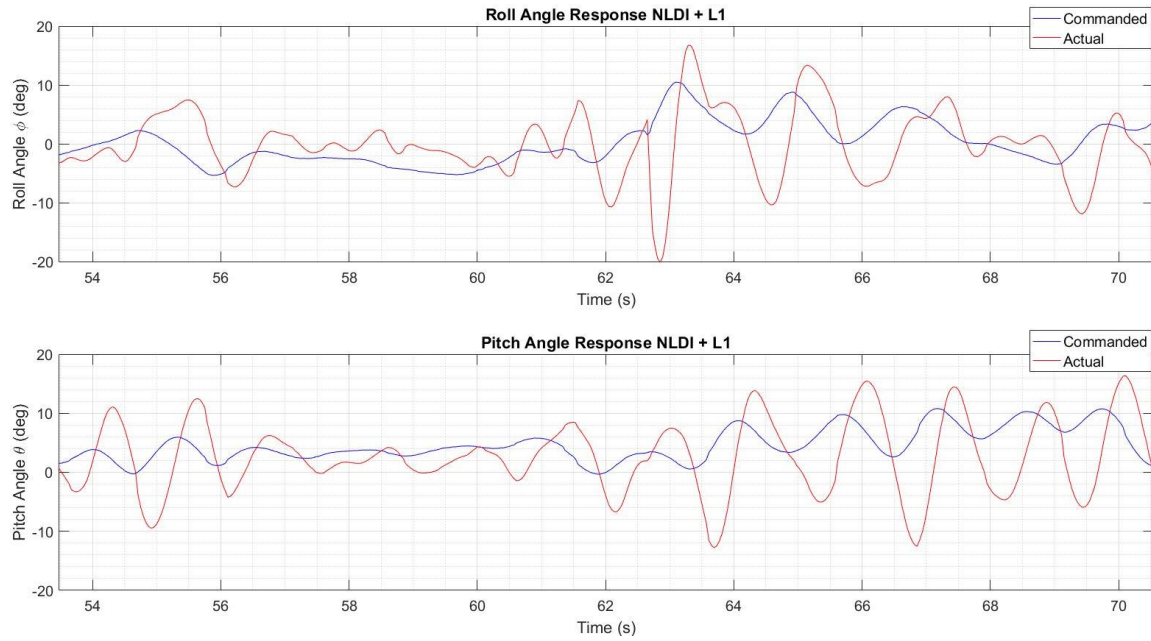


Figure 7.17. Roll and pitch angle response for baseline NLDI with \mathcal{L}_1 adaptive control augmentation under medium wind conditions.

7.2.4. High wind disturbance conditions. Table 7.5 shows the performance index comparison for high wind conditions for the three controllers together with the average and maximum wind speed peak for each flight. For high wind conditions, which can be characterized by wind speeds over 5kts, all three controllers are not able to sustain stable and controlled flight, resulting in crashes.

Table 7.5

Performance Index Comparison for High Wind Conditions

Controller	PM Attitude	PM Rates	PM Trajectory	PM PWM	Global PI	Flight Time (s)	Average Wind Speed (kts)	Maximum Wind Peak (kts)
Baseline NLDI	0.4570	0.0000	0.4290	0.729532	0.1535	29.31	5.0	8.3
NLDI + NN	0.4454	0.0000	0.3799	0.726492	0.0667	13.62	5.4	8.8
NLDI + L1	0.6379	0.0000	0.0822	0.728157	0.2577	17.65	6.5	8.5

Figure 7.18 shows the 3D trajectory of the Flamewheel quadcopter with the baseline NLDI, baseline NLDI with ANN augmentation and baseline NLDI with \mathcal{L}_1 augmentation during the autonomous altitude and position hold outdoors in the presence of high wind disturbance conditions. As it is shown, all three flights resulted in crashes, indicating that for high wind conditions even the addition of ANN and \mathcal{L}_1 augmentation will not be able to stabilize the relatively light-weight Flamewheel in flight.

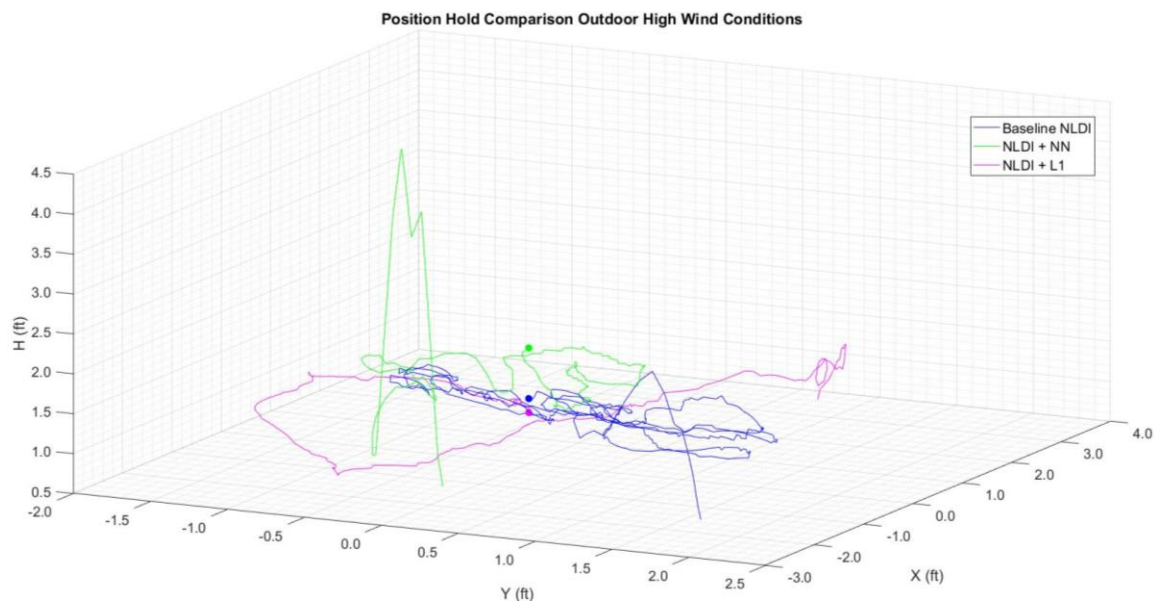


Figure 7.18. 3D trajectory comparison for high wind conditions.

Adaptive ANN augmentation. Figure 7.19 shows the outputs of the adaptive ANN at high wind conditions with $\Gamma_1 = \Gamma_2 = 31.4$ to attempt for fast adaptation of the network parameters to the high wind disturbance. Although the network is trying to adjust the internal parameters to correct for the wind effects, the large wind disturbance is too significant for the adaptive ANN to correct.

\mathcal{L}_1 output-feedback adaptive control augmentation. Figure 7.20 shows the outputs of the \mathcal{L}_1 adaptive controller for high wind conditions with an output multiplier equal to 3.4 to attempt to augment values to correct for the high wind disturbance. However, similar to the adaptive ANN, the large magnitude wind disturbance is too significant to correct for by the \mathcal{L}_1 adaptive controller.

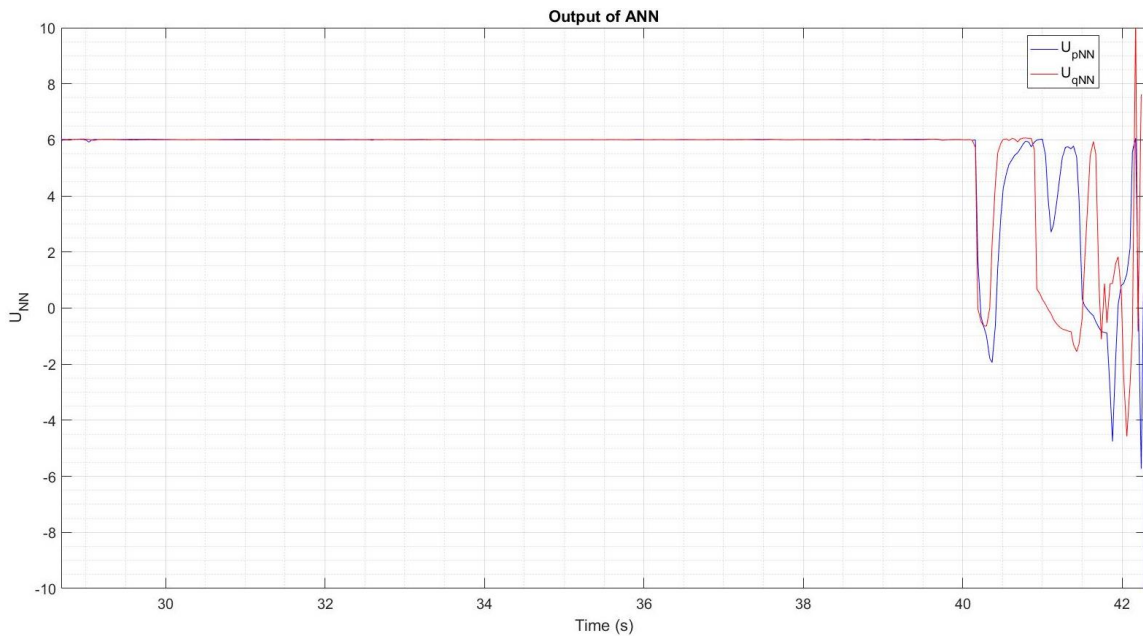


Figure 7.19. Outputs of the adaptive ANN for high wind conditions.

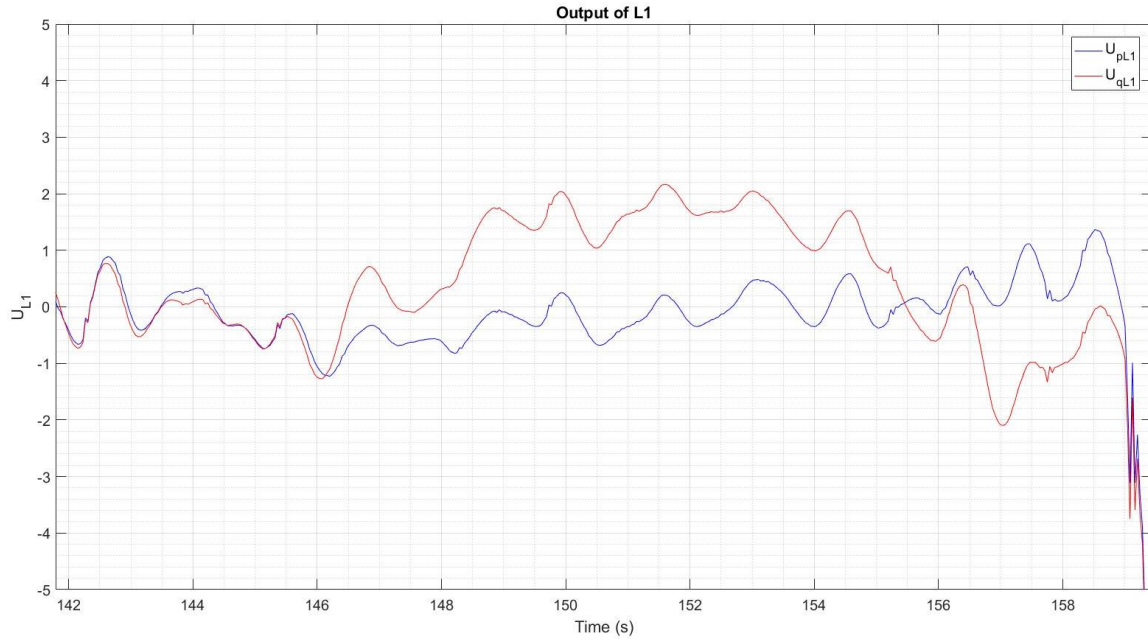


Figure 7.20. Outputs of the \mathcal{L}_1 adaptive controller for high wind conditions.

8. Conclusions

This thesis focused on the design, development and implementation of robust control algorithms for disturbance rejection in rotorcraft UAVs with the focus on the rejection of external disturbances caused by wind influences.

An evaluation of the performance of the baseline nonlinear dynamic inversion (NLDI) control law, the wind rejection extension for the NLDI, the NLDI with adaptive artificial neural networks (ANN) augmentation and the NLDI with \mathcal{L}_1 output-feedback adaptive control augmentation in a simulation environment was performed. Results for a Monte Carlo analysis throughout a realistic wind envelope using the four considered controllers showed that the extended NLDI provides the best position tracking capabilities but sacrifices rotational stability. The NLDI controllers with adaptive ANN and \mathcal{L}_1 augmentation outperform the baseline NLDI in rotational dynamic stability and position tracking capabilities.

An evaluation of the performance of the baseline NLDI, NLDI with adaptive ANN augmentation and NLDI with \mathcal{L}_1 output-feedback adaptive control augmentation implemented in a DJI Flamewheel F330 quadrotor was performed in flight testing in both indoor nominal conditions, outdoor nominal conditions and outdoor disturbed conditions. Results showed that all three controllers behave uniformly in both indoor and outdoor nominal conditions. The NLDI with adaptive ANN and \mathcal{L}_1 augmentation outperformed the baseline NLDI in all performance metrics when the magnitude of the wind disturbance increased. No implementation was performed for the extended NLDI due to decreased robustness as compared to the baseline NLDI shown in the simulation results and because of the additional required wind estimation/measurement method to supply wind velocities and accelerations.

It can therefore be concluded that the augmentation of the baseline NLDI control law with adaptive ANN and \mathcal{L}_1 output-feedback adaptive control improves the robustness of the translational and rotational dynamics of a rotorcraft UAV in the presence of wind disturbances.

Although no results are shown in this thesis, the conclusion can be extended to state that the addition of adaptive ANN and \mathcal{L}_1 output-feedback adaptive control augmentation will increase the robustness of a baseline NLDI control law applied to a rotorcraft in the presence of any type of disturbances or uncertainties. Unlike the extended NLDI, the adaptive ANN and the \mathcal{L}_1 do not require explicit information about the type of disturbance. In this thesis, the considered inversion linearization errors in the NLDI control law were caused by adverse external wind disturbances. However, these inversion errors can be considered to be the result of any type of external or internal

disturbances and uncertainties.

The effect of the computational power limitations of the PX1 and the resulting relatively slow sampling frequency was observed to be significant on the performance of both the adaptive controllers in flight. The Nyquist frequency should ideally be a few orders of magnitude larger, preferably larger than five, than the highest dynamic frequency found in the system. During test flights the Nyquist frequency was observed to be slightly more than two times the highest dynamic frequency in the system. Although both the adaptive controllers showed a stabilizing performance, their robustness against higher frequency disturbances can be greatly improved by increasing the computational power of the onboard flight computer.

Although the augmentation of the baseline NLDI control law with adaptive ANN and \mathcal{L}_1 was shown to add robustness against external wind disturbances, an essential requirement is that the baseline control law is tuned and stable. Without this prerequisite, the robustness and stability of the rotorcraft system will be degraded even under nominal conditions.

9. Future Work & Recommendations

The research presented in this thesis can be extended and improved in multiple different ways.

Firstly, a wind estimation/measurement method can be developed to allow for an estimation of the wind velocity and acceleration to be available. This estimation can subsequently be used to implement the extended NLDI control law for hardware-in-the-loop flight testing to evaluate the performance. Much detail must be placed in the development of an accurate wind estimation/measurement method since simulation

results showed that the performance of the extended NLDI is very sensitive to the accuracy and precision of the wind velocity and acceleration estimates.

Additionally, the PX1 runs a soft real-time operating system with a sampling frequency of around 25Hz. To improve the performance of the discretized \mathcal{L}_1 controller and the adaptive ANN to allow for higher frequency dynamic characteristics, the PX1 can be replaced by a faster computer equipped with a hard real-time operating system and more computational power in order to increase the sampling frequency. The PC104 Advantech PCM computer would be a potential candidate for this application.

A more in depth analysis can be performed on the adaptive ANN and \mathcal{L}_1 controllers to evaluate if the additional tuning of the numerous parameters can improve the performance of both controllers in simulation and implementation.

More flight tests should be performed to obtain a more complete set of implementation results shown in Chapter 6. Inconsistencies were occasionally encountered in the outdoor flight testing campaign. To decrease the effect of outliers, more flight tests should be performed for each controller to give a more complete image of the performance of the controllers exposed to external wind disturbances.

Improvements can be made to the characterization of the wind by the ground weather station. The wind environment seemed to be varying locally on the artificial turf softball field and since the Flamewheel was flown at safe distance from the ground weather station, some error might be introduced in the wind characterization. The local wind disturbances as felt by the Flamewheel in flight could be quite different from the local wind measured by the ground weather station. To improve the characterization of the wind, multiple ground weather stations could be placed around the artificial turf

softball field to obtain a sense of varying wind conditions across the field during Flamewheel flight.

A more complete set of internal and external disturbances and uncertainties could be considered for which a performance analysis can be performed using the baseline NLDI, NLDI with adaptive ANN augmentation and NLDI with \mathcal{L}_1 augmentation control laws. Possible disturbances and uncertainties that can be considered include motor saturation, sensor malfunction, the addition of extra weight to a tuned baseline NLDI and hardware failures in the form of a partially broken propeller.

Lastly, this thesis considered two types of adaptive control; model reference based adaptive control and intelligent control. However, more controllers including those discussed in the literature review in Chapter 2 can be considered and evaluated under external disturbance conditions.

REFERENCES

- Abeywardena, D., Wang, Z., Dissanayake, G., Waslander, S. L., & Kodagoda, S. (2014). Model-aided State Estimation for Quadrotor Micro Air Vehicles Amidst Wind Disturbances. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL.
- ACRONAME. (2019). *LightWare SF11 (50 or 120 meter)*. Retrieved from acroname.com: <https://acroname.com/store/r431-sf11b-50?sku=R431-SF11B-50>
- Al-Mahasneh, J., Anavatti, S., & Garatt, M. (2017). The Development of Neural Networks applications from Perceptron to Deep Learning. *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*, (pp. 1-6). Surabaya.
- Amazon. (2019). *Hobbypower SimonK 30A ESC Brushless Speed Controller BEC 2A for Quadcopter F450 X525 (Pack of 4 pcs)*. Retrieved from amazon.com: https://www.amazon.com/Hobbypower-SimonK-Brushless-Controller-Quadcopter/dp/B00QRR7N32/ref=pd_bxgy_21_img_2/138-3620583-8527503?_encoding=UTF8&pd_rd_i=B00QRR7N32&pd_rd_r=5de64f6e-63ac-11e9-a637-6b060766031a&pd_rd_w=5YAxj&pd_rd_wg=rAp4M&pf_rd_p=a2006322-0b
- Amazon. (2019). *WOAFLY 2212 920kv Brushless Motor(CW/CCW)-4 Pcs For DJI Phantom F330 F450 F550 X525*. Retrieved from amazon.com: <https://www.amazon.com/WOAFLY-920kv-Brushless-Motor-Phantom/dp/B01HM8EXJG>
- ArduPilot Dev Team. (2019). *UBlox GPS + Compass Module*. Retrieved from ardupilot.org: <http://ardupilot.org/copter/docs/common-installing-3dr-ublox-gps-compass-module.html>
- Astrom, K. J., & Murray, R. (2012). *Feedback Systems*. Princeton: Princeton University Press.
- Ataka, A., Tnunay, H., Inovan, R., Abdurrohman, M., Preastianto, H., Cahyadi, A., & Yamamoto, Y. (2013). Controllability and Observability Analysis of the Gain Scheduling Based Linearization for UAV Quadrotor. *2013 International Conference on Robotics, Biomimetrics, Intelligent Computational Systems* (pp. 212-218). Yogyakarta: IEEE.
- Bangura, M. (2017). *Aerodynamics and Control of Quadrotors*. Canberra: The Australian National University.
- Bangura, M., & Mahony, R. (2012). Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor. *Proceedings of Australasian Conference on Robotics and*

- Automation*. Wellington, New Zealand: Victoria University of Wellington.
- Bannwarth, J., Chen, Z., Stol, K., & MacDonald, B. (2016). Disturbance Accommodation Control for Wind Rejection of a Quadcopter. *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 695-701). Arlington.
- Beard, R., & McLain, T. (2012). *Small Unmanned Aircraft: Theory and Practice*. Princeton: Princeton University Press.
- Bo, G., Xin, L., Hui, Z., & Ling, W. (2016). Quadrotor Helicopter Attitude Control using Cascade PID. *2016 Chinese Control and Decision Conference* (pp. 5158-5163). Yinchuan: IEEE.
- Bouabdalla, S., & Siegwart, R. (2005). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (pp. 2247-2252). Barcelona: IEEE.
- Bros, P. (2017). *ULTIMETER 2100 Weather Station*. Retrieved from peetbros.com: <http://www.peetbros.com/shop/item.aspx?itemid=2>
- Burken, J. J., Williams-Hayes, P., Kaneshige, J. T., & Stachowiak, S. J. (2006). Adaptive Control Using Neural Network Augmentation for a Modified F-15 Aircraft. *2006 14th Mediterranean Conference on Control and Automation*. Ancona.
- Calise, A., Lee, S., & Sharma, M. (2001). Development of a Reconfigurable Flight Control Law for Tailles Aircraft. *Journal of Guidance, Control, and Dynamics*, 896-902.
- Calise, A., Sharma, M., & Corban, J. (2000). Adaptive Autopilot Design for Guided Munitions. *Journal of Guidance, Control, and Dynamics*, 837-843.
- Campa, G., Fravolini, M., & Napolitano, M. (2002). A Library of Adaptive Neural Networks for Control Purposes. *2002 IEEE International Symposium on Computer Aided Control System Design Proceedings* (pp. 115-120). Glasgow: IEEE.
- Campa, G., Fravolini, M., Mammarella, M., & Napolitano, M. (2011). Bounding Set Calculation for Neural Network-Based Output Feedback Adaptive Control Systems. *Neural Computing and Applications*, 373-387.
- Campa, G., Gu, Y., Seanor, B., Napolitano, M., Pollini, L., & Fravolini, M. (2007). Design and Flight-Testing of Non-Linear Formation Control Laws. *Elsevier Control Engineering Practice*, 1077-1092.
- Cao, C., & Hovakimyan, N. (2006a). Design and Analysis of a Novel L1 Adaptive

- Control Architecture Part I: Control Signal and Asymptotic Stability. *American Control Conference*, (pp. 3397-3402). Minneapolis.
- Cao, C., & Hovakimyan, N. (2006b). Design and Analysis of a Novel L1 Adaptive Control Architecture, Part II: Guaranteed transient performance. *American Control Conference*, (pp. 3403-3408). Minneapolis.
- Cao, C., & Hovakimyan, N. (2008). L1 Adaptive Output Feedback Controller for Non Strictly Positive Real Reference Systems with Applications to Aerospace Examples. *AIAA Guidance, Navigation and Control Conference and Exhibit* (pp. 1-16). Honolulu: AIAA.
- Cao, C., & Hovakimyan, N. (2009). L1 Adaptive Output-Feedback Controller for Non-Strictly-Positive-Real Reference Systems: Missile Longitudinal autopilot Design. *Journal of Guidance, Control, and Dynamics*, 717-726.
- Cao, C., Patel, V., Reddy, K., & Hovakimyan, N. (2006). Are Phase and Time-delay Margins Always Adversely Affected by High-Gain? *AIAA Guidance, Navigation, and Control Conference and Exhibit* (pp. 1-15). Keystone: AIAA.
- Cao, N., & Lynch, A. F. (2016). Inner-Outer Loop Control for Quadrotor UAVs with Input and State Constraints. *IEEE Transactions on Control Systems Technology* (pp. 1797-1804). IEEE.
- CBS News. (2014). *FAA criminalizes flying drones near certain locations*. Retrieved February 25, 2019, from cbsnews.com: <https://www.cbsnews.com/news/faa-criminalizes-flying-drones-near-certain-locations/>
- CBS News. (2018). *Children on remote island receive first drone-delivered vaccines*. Retrieved February 25, 2019, from cbsnews.com: <https://www.cbsnews.com/news/first-drone-delivered-vaccines-for-children-on-remote-island-vanuatu/>
- Chen, M., & Huzmezan, M. (2003). A Combined MBPC/2 DOF H_∞ Controller for a quadrotor UAV. *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Austin: AIAA.
- Cohn, P., Green, A., Langstaff, M., & Roller, M. (2017). *Commercial drones are here: the future of unmanned aerial systems*. Retrieved February 25, 2019, from mckinsey.com: <https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems>
- Coulter, N. (2018). *Design of an Attitude Control System for a Spacecraft with Propellant Sloss Dynamics*. Daytona Beach: Embry-Riddle Aeronautical University (Master's thesis).

- Coulter, N., Moncayo, H., & Engblom, W. (2018). Evaluation of Sailing Flight Optimization Algorithm for The Dual Aircraft Platform Concept. *2018 Modeling and Simulation Technologies Conference*. Atlanta: AIAA.
- Darack, E. (2011, 5 17). *A Brief History of Unmanned Aircraft*. (Air&Space Smithsonian) Retrieved February 25, 2019, from airspacemag.com: <https://www.airspacemag.com/photos/a-brief-history-of-unmanned-aircraft-174072843/?page=9>
- Desjardins, J. (2018). *Amazon and UPS are betting big on drone delivery*. Retrieved February 25, 2019, from businessinsider.com: <https://www.businessinsider.com/amazon-and-ups-are-betting-big-on-drone-delivery-2018-3>
- Dong, Y., Fu, J., Yu, B., Zhang, Y., & Al, J. (2015). Position and Heading Angle Control of an Unmanned Quadrotor Helicopter Using LQR Method. *Proceedings of the 34th Chinese Control Conference* (pp. 5566-5571). Hangzhou: IEEE.
- Dydek, Z., Annaswamy, A., & Lavretsky, E. (2013). Adaptive Control of Quadrotor UAVs: A Design Trade Study with Flight Evaluations. *IEEE Transactions on Control Systems Technology*, 1400-1406.
- Escareno, J., Salazar, S., Romero, H., & Lozano, R. (2013). Trajectory Control of a Quadrotor Subject to 2D Wind Disturbances. *Journal of Intelligent & Robotic Systems*, 51-63.
- Etele, J. (2006). *Overview of Wind Gust Modelling with Application to Autonomous Low-Level UAV Control*. Ottawa: Defence Research and Development Canada.
- Evans, M. (2018). *Gatwick airport drone chaos: man, 47, and woman, 54, arrested in Crawley*. Retrieved February 25, 2019, from telegraph.co.uk: <https://www.telegraph.co.uk/news/2018/12/22/gatwick-airport-drone-chaos-man-woman-arrested-passengers-brace/>
- FAA. (2017). *FAA Releases Updated Drone Sighting Reports*. Retrieved February 25, 2019, from faa.gov: <https://www.faa.gov/news/updates/?newsId=87565>
- FAA. (2018). *FAA Releases Aerospace Forecast*. Retrieved February 25, 2019, from faa.gov: <https://www.faa.gov/news/updates/?newsId=89870>
- Falkenberg, O., Witt, J., Weltin, U., & Werner, H. (2012). Model Identification and H ∞ Attitude Control for Quadrotor MAV's. *International Conference on Intelligent Robotics and Applications*, (pp. 460-471). Montreal.
- Gallagher, S. (2013). *German chancellor's drone "attack" shows the threat of weaponized UAVs*. Retrieved February 25, 2019, from arstechnica.com:

<https://arstechnica.com/information-technology/2013/09/german-chancellors-drone-attack-shows-the-threat-of-weaponized-uavs/>

- Garcia, D. F. (2017). *Design Development and Implementation of Intelligent Algorithms to Increase Autonomy of Quadrotor Unmanned Missions*. Daytona Beach: Embry-Riddle Aeronautical University (Master's thesis).
- Garcia, D., Moncayo, H., Perez, A., & Jain, C. (2016). Low Cost Implementation of a Biometric Approach for UAV Health Monitoring. *American Control Conference (ACC)*. Boston.
- Garcia, D., Perez, A., Moncayo, H., & Rivera, K. (2017). Spacecraft Health Monitoring Using a Biomimetic Fault Diagnosis Scheme. *AIAA Information Systems-AIAA Infotech*. Grapevine.
- Garcia, D., Perez, A., Moncayo, H., Rivera, K., DuPuis, M., & Mueller, R. (2018). Spacecraft Health Monitoring Using a Biomimetic Fault Diagnosis Scheme. *AIAA Journal of Aerospace Information Systems*, 396-413.
- Garcia, R. A., Rubio, F. R., & Ortega, M. G. (2012). Robust PID Control of the Quadrotor Helicopter. *2nd IFAC Conference on Advances in PID Control* (pp. 229-234). Elsevier.
- Geiser, M., Xargay, E., & Hovakimyan, N. (2011). L1 Adaptive Augmented Dynamic Inversion Controller for a High Agility UAV. *AIAA Guidance, Navigation, and Control Conference* (pp. 1-18). Portland: AIAA.
- Glaser, A. (2017). *Police departments are using drones to find and chase down suspects*. Retrieved February 25, 2019, from recode.net: <https://www.recode.net/2017/4/6/15209290/police-fire-department-acquired-drone-us-flying-robot-law-enforcement>
- Google. (2019). *Google Earth*. Retrieved from google.com/earth: <https://www.google.com/earth/>
- Harkegard, O. (2001). *Flight Control Design using Backstepping*. Linkoping: Linkopings Universitet (Master's thesis).
- Hoffmann, G., Huang, H., Waslander, S., & Tomlin, C. (2007). Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment. *AIAA Guidance, Navigation and Control Conference and Exhibit* (pp. 1-20). Hilton Head: AIAA.
- Hoffmann, G., Huang, H., Waslander, S., & Tomlin, C. (2009). Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering. *2009 IEEE International Conference on Robotics and Automation* (pp. 3277-3282). Kobe: IEEE.

- Horizon Hobby. (2019). *DX8 8-Channel DSMX® Transmitter Only, Mode 2 (SPMR8810)*. Retrieved from horizonhobby.com:
<https://www.horizonhobby.com/dx8-8-channel-dsmx-reg%3B-transmitter--spm8810>
- Hovakimyan, N., & Cao, C. (2010). *L1 Adaptive Control Theory; Guaranteed Robustness with Fast Adaptation*. Philadelphia: Society for Industrial and Applied Mathematics.
- Huang, H., Hoffmann, G. M., Waslander, S. L., & Tomlin, C. J. (2009). Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering. *IEEE International Conference on Robotics and Automation*. Kobe, Japan.
- Huo, X., Huo, M., & Karimi, H. (2014). Attitude Stabilization Control of a Quadrotor UAV by Usig Backstepping Approach. *Mathematical Problems in Engineering*, 1-9.
- Ireland, M., Vargas, A., & Anderson, D. (2015). A Comparison of Closed-Loop Performance of Multirotor Configurations Using Non-Linear Dynamic Inversion Control. *Aerospace*, 325-352.
- Ito, D., Georgie, J., Valasek, J., & Ward, D. (2002). *Reentry Vehicle Flight Controls Design Guidelines: Dynamic Inversion*. Houston: NASA.
- Jafarnejadsani, H., Lee, H., & Hovakimyan, N. (2017). An L1 Adaptive Control Design for Output-Feedback Sampled-Data Systems. *2017 American Control Conference*, (pp. 5744-5749). Seattle.
- Kaneshige, J., Bull, J., & Totah, J. (2000). Generic Neural Flight Control and Autopilot System. *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Denver: AIAA.
- Kaufman, L., & Somaiya, R. (2013). *Drones offer journalists a wider view*. Retrieved February 25, 2019, from nytimes.com:
<https://www.nytimes.com/2013/11/25/business/media/drones-offer-journalists-a-wider-view.html>
- Khalil, H. K. (2002). *Nonlinear Systems Third Edition*. Upper Saddle River: Prentice Hall.
- Kim, D. (2018). Fuzzy Model-Based Control of a Quadrotor. *Elsevier Fuzzy Sets and Systems*, 1-12.
- Krishna, L., & Murphy, R. R. (2017). A Review on Cybersecurity Vulnerabilities for Unmanned Aerial Vehicles. *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. Shanghai.

- Kwakernaak, H. (1993). Robust Control and H_∞ Optimization Tutorial Paper. *Automatica*, 225-273.
- Langelaan, J., Alley, N., & Neidhoefer, J. (2011). Wind Field Estimation for Small Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4), 1016-1030.
- Lavrinc, D. (2012). *Forget the helicopter: new drone cuts cost of aerial video*. Retrieved February 25, 2019, from wired.com: <https://www.wired.com/2012/05/drone-auto-vids/>
- Lee, D., Kim, H., & Sastry, S. (2009). Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter. *International Journal of Control, Automation, and Systems*, 419-428.
- Leishman, J. G. (2000). *Principles of Helicopter Aerodynamics*. Cambridge: Cambridge University Press.
- Leman, T., Xargay, E., Dullerud, G., & Hovakimyan, N. (2009). L1 Adaptive Control Augmentation System for the X-48B Aircraft. *AIAA Guidance, Navigation, and Control Conference* (pp. 1-14). Chicago: AIAA.
- Leman, T., Xargay, E., Dullerud, G., Hovakimyan, N., & Wendel, T. (2009). L1 Adaptive Control Augmentation System for the X-48B Aircraft. *AIAA Guidance, Navigation, and Control Conference* (pp. 1-14). Chicago: AIAA.
- Levine, W. S. (2011). *The Control Handbook; control system advanced methods*. Boca Raton: CRC Press.
- Lewis, F., Yesildirek, A., & Liu, K. (1996). Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance. *IEEE Transactions on Neural Networks*, 388-399.
- Li, J., & Li, Y. (2011). Dynamic Analysis and PID Control for a Quadrotor. *International Conference on Mechatronics and Automation* (pp. 573-578). Beijing: IEEE.
- Lopez, J., Dormido, R., Dormido, S., & Gomez, J. P. (2015). A Robust H-infinity Controller for an UAV Flight Control System. *The Scientific World Journal*, 10(1155), 1-11.
- Lu, Y., Sundararajan, N., & Saratchandran, P. (2000). Analysis of Minimal Radial Basis Function Network Algorithm for Real-time Identification of Nonlinear Dynamic Systems. *IEEE Proceedings on Control Theory and Application*, 476.
- Madani, T., & Benallegue, A. (2006). Backstepping Control for a Quadrotor Helicopter. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent*

Robots and Systems (pp. 3255-3260). Beijing: IEEE.

- Madrigal, A. C. (2011). *Inside the Drone Missions to Fukushima*. Retrieved February 25, 2019, from theatlantic.com:
<https://www.theatlantic.com/technology/archive/2011/04/inside-the-drone-missions-to-fukushima/237981/#slide10>
- McFarland, M. B., & Calise, A. J. (2000). Multilayer Neural Networks and Adaptive Nonlinear Control of Agile Anti-air Missiles. *IEEE Transactions on Control Systems Technology*.
- McGivering, J. (2012). *Drones to protect Nepal's endangered species from poachers*. Retrieved February 25, 2019, from bbc.com: <https://www.bbc.com/news/science-environment-18527119>
- Moncayo, H., & Perhinschi, M. (2011). *Aircraft Fault Tolerance: A Biologically Inspired Immune Framework for Sub-System Failures*. Saarbrücken VDM Verlag Dr. Müller.
- Moncayo, H., Krishnamoorthy, K., Wilburn, B., Wilburn, J., Perhinschi, M. G., & Lyons, B. (2013). Performance Analysis of Fault Tolerant UAV Baseline Control Laws with L1 Adaptive Augmentation. *Journal of Modeling, Simulation, Identification, and Control*, 1(4), 137-163.
- Moncayo, H., Perhinschi, M., Wilburn, B., Wilburn, J., & Karas, O. (2012). Extended Nonlinear Dynamic Inversion Control Laws for Unmanned Air Vehicles. *AIAA Guidance, Navigation, and Control Conference* (pp. 1-17). Minneapolis: AIAA.
- Napolitano, M. (2012). *Aircraft Dynamics From Modeling to Simulation*. Hoboken, NJ: Wiley.
- Narendra, K., & Annaswamy, A. (2005). *Stable Adaptive Systems*. New Jersey: Prentice Hall.
- NASA. (2019). *Unmanned Aircraft Systems (UAS) Traffic Management (UTM)*. Retrieved February 25, 2019, from nasa.gov: <https://utm.arc.nasa.gov/index.shtml>
- Nshuti, C. (2017). *Design of Flight Control Laws for a Novel Stratospheric Dual-Aircraft Platform*. Daytona Beach: Embry-Riddle Aeronautical University (Master's thesis).
- Ogata, K. (1995). *Discrete-Time Control Systems*. Englewood Cliffs: Prentice-Hall.
- Omari, S., Hua, M.-D. D., & Hamel, T. (2013). Nonlinear Control of VTOL UAVs Incorporating Flapping Dynamics. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2419-2425). Tokyo, Japan: IEEE/RSJ.

- Palomaki, R. T., Rose, N. T., Van den Bossche, M., Sherman, T. J., & De Wekker, S. F. (2017). Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft. *Journal of Atmospheric and Oceanic Technology*, 34(5), 1183-1191.
- Pan, K., Chen, Y., Wang, Z., Wu, H., & Cheng, L. (2018). Quadrotor Control based on Self-Tuning LQR. *Proceedings of the 37th Chinese Control Conference* (pp. 9974-9979). Wuhan: IEEE.
- Passino, K., & Yurkovich, S. (1997). *Fuzzy Control*. Menlo Park: Addison Wesley Longman, Inc.
- Pegues, J. (2015). *Drone over White House highlights security concerns*. Retrieved February 25, 2019, from <https://www.cbsnews.com/news/drone-over-white-house-sparks-new-security-concerns/>
- Perez, A. (2016). *Development of Fault Tolerant Adaptive Control Laws for Aerospace Systems*. Daytona Beach: Embry-Riddle Aeronautical University (Phd dissertation).
- Perez, A. A., Moncayo, H., Prazenica, R. J., Zacny, K., Mueller, R. P., Ebert, T., & DuPuis, M. (2016). Control Laws Development for a Free-Flying Unmanned Robotic System to Support Interplanetary Bodies Prospecting and Characterization Missions. *AIAA Guidance, Navigation, and Control Conference*. San Diego.
- Perhinschi, M. G., Burken, J., & Campa, G. (2006). Comparison of Different Neural Augmentations for the Fault Tolerant Control Laws of the WVU YF-22 Model Aircraft. *2006 14th Mediterranean Conference on Control and Automation*. Ancona.
- Perhinschi, M., Moncayo, H., Wilburn, B., Wilburn, J., Karas, O., & Bartlett, A. (2014). Neurally-augmented immunity-based detection and identification of aircraft subsystem failures. *The Aeronautical Journal*, 118(1205), 1-22.
- Perhinschi, M., Napolitano, M., Campa, G., Seanor, B., Gururajan, S., & Yu, G. (2005). Design and Flight Testing of Intelligent Flight Control Laws for the WVU YF-22 Model Aircraft. *AIAA Guidance, Navigation, and Control Conference and Exhibit* (pp. 1-12). San Francisco: AIAA.
- Pilot Engineering Group Mathworks. (2017). *Pixhawk Pilot Support Package (PSP) User Guide*. Mathworks.
- Propellers, A. (2019). *B8x4.5MR-B4*. Retrieved from www.apcprop.com: <http://www.apcprop.com/product/b8x4-5mr-b4/>
- Px4 Dev Team. (2019, 4 4). *Pixhawk 1 Flight Controller*. Retrieved from

[www.docs.px4.io: https://docs.px4.io/en/flight_controller/pixhawk.html](https://docs.px4.io/en/flight_controller/pixhawk.html)

Rauw, M. (2001). *FDC 1.2 - A SIMULINK Toolbox for Flight Dynamics and Control Analysis*. dutchroll.com.

RCJuice. (2019). *HobbyStar 3000mAh 11.1V, 3S 30C LiPo Battery*. Retrieved from [www.rcjuice.com: https://www.rcjuice.com/hobbystar-3000mah-11-1v-3s-30c-lipo-battery.html](https://www.rcjuice.com/hobbystar-3000mah-11-1v-3s-30c-lipo-battery.html)

Rivera, K. (2018). *Design and Implementation of Intelligent Guidance Algorithms for UAV Mission Protection*. Daytona Beach: Embry-Riddle Aeronautical University (Master's thesis).

Rivera, K. P., Moncayo, H., Verberne, J., & Festa, D. (2019). Design and Implementation of Intelligent Decision-making Algorithms for Unmanned Aerial Vehicles Mission Protection. *AIAA SciTech 2019 Forum*. San Diego.

Rodriguez, A. F., Andersen, E., Bradley, J., & Taylor, C. (2007). Wind Estimation Using an Optical Flow Sensor on a Miniature Air Vehicle. *AIAA Guidance, Navigation and Control Conference and Exhibit*. Hilton Head, SC.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Representations by Back-Propagating Errors. *International Journal of Science Nature*, 533-536.

Rysdyk, R., & Calise, A. (1998). Fault Tolerant Flight Control via Adaptive Neural Network Augmentation. *Guidance, Navigation, and Control Conference and Exhibit* (pp. 1722-1728). Boston: AIAA.

Salih, A. L., Moghavvemi, M., Mohamed, H. A., & Sallom Gaeid, K. (2010). Flight PID controller design for a UAV quadrotor. *Scientific Research and Essays, Academic Journals*, 3660-3667.

Samy, I., Fan, I., & Perinpanayagam, S. (2010). Fault Diagnosis of Rolling Element Bearings using an EMRAN RBF Neural Network - Demonstrated using Real Experimental Data. *2010 Sixth International Conference on Natural Computation (ICNC 2010)*, (pp. 287-291). Yantai.

Sharma, M., & Calise, A. (2005). Neural-Network Augmentation of Existing Linear Controllers. *Journal of Guidance, Control, and Dynamics*, 12-19.

Sikkel, L., De Croon, G., De Wagter, C., & Chu, Q. (2016). A Novel Online Model-Based Wind Estimation Approach for Quadrotor Micro Air Vehicles using Low Cost MEMS IMUs. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2141-2146). Daejeon: IEEE.

Simplicio, P., Pavel, M., van Kampen, E., & Chu, Q. (2013). An Acceleration

Measurements-Based Approach for Helicopter Nonlinear Flight Control using Incremental Nonlinear Dynamic Inversion. *Elsevier Control Engineering Practice*, 1065-1077.

Slotine, J., & Li, W. (1991). *Applied Non-Linear Control*. New Jersey: Prentice Hall.

Smeur, E., de Croon, G., & Chu, Q. (2016). Cascaded Incremental Nonlinear Dynamic Inversion Control for MAV Disturbance Rejection. *2016 IEEE/RSJ International Conference on intelligent Robotics and Systems (IROS)*, (pp. 1-12). Daejeon.

SparkFun Electronics. (2013). *MPU-6000 - Motion Processing Unit*. Retrieved from www.sparkfun.com: <https://www.sparkfun.com/products/retired/11234>

Suarez Fernandez, R., Dominguez, S., & Campoy, P. (2017). L1 Adaptive Control for Wind Gust Rejection in Quad-rotor UAV Wind Turbine Inspection. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 1840-1849). Miami.

Sydney, N., Smyth, B., & Paley, D. A. (2013). Dynamic Control of Autonomous Quadrotor Flight in an Estimated Wind Field. *52nd IEEE Conference on Decision and Control*. Florence, Italy.

Tomic, T., Schmid, K., Lutz, P., Mathers, A., & Haddadin, S. (2016). The Flying Anemometer: Unified Estimation of Wind Velocity from Aerodynamic Power and Wrenches. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1637-1644). Daejeon: IEEE.

Verberne, J., & Moncayo, H. (2019). Comparison of Adaptive Control Laws for Wind Rejection in Quadrotor UAVs. *ASME 2019 Dynamic Systems and Control Conference (DSCC)*. Park City, Utah: ASME.

Verberne, J., & Moncayo, H. (2019). Robust Control Architecture for Wind Rejection in Quadrotors. *The 2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, Georgia: ICUAS.

Verberne, J., Betancur, Y., Riverak, K., Coulter, N., & Moncayo, H. (2019). Comparison of MRAC and L1 Adaptive Controllers for a Gimballed Mini-Free Flyer. *AIAA SciTech 2019 Forum*. San Diego, California: AIAA.

Wang, L., He, Y., Zhang, Z., & He, C. (2013). Trajectory Tracking of Quadrotor Aerial Robot using Improved Dynamic Inversion Method. *Computer Science & Communications*, 343-348.

Wang, P., Man, Z., Cao, Z., Zheng, J., & Zhao, Y. (2016). Dynamics Modelling and Linear Control of Quadcopter. *2016 International Conference on Advanced Mechatronic Systems* (pp. 498-503). Melbourne, Australia: IEEE.

- Wang, X., Kharisov, E., & Hovakimyan, N. (2014). Real-Time L1 Adaptive Control for Uncertain Networked Control Systems. *IEEE Transactions on Automatic Control*, 1-32.
- Waslander, S. L., & Wang, C. (2009). Wind Disturbance Estimation and Rejection for Quadrotor Position Control. *AIAA Unlimited conference and Exhibit*. Seattle, WA.
- Weil, M. (2013). *Drone crashes into Virginia bull run crowd*. Retrieved February 25, 2019, from washingtonpost.com: https://www.washingtonpost.com/local/drone-crashes-into-virginia-bull-run-crowd/2013/08/26/424e0b9e-0e00-11e3-85b6-d27422650fd5_story.html?noredirect=on&utm_term=.07e9c0cb2e67
- Whitaker, H., Yamron, J., & Kezer, A. (1958). *Design of Model Reference Adaptive Control Systems for Aircraft*. Cambridge: Instrumentation Laboratory, MIT.
- Whitwam, R. (2016). *NASA testing helicopter drone to accompany next Mars rover*. Retrieved February 25, 2019, from extremetech.com: <http://www.extremetech.com/extreme/229937-nasa-testing-helicopter-drone-to-accompany-next-mars-rover>
- Xu, R., & Ozguner, U. (2006). Sliding Mode Control of a Quadrotor Helicopter. *Proceedings of the 45th IEEE Conference on Decision & Control* (pp. 4957-4962). San Diego: IEEE.
- Yeo, D. W., Sydney, N., & Paley, D. A. (2015). Onboard Flow Sensing for Downwash Detection and Avoidance with a Small Quadrotor Helicopter. *AIAA Guidance, Navigation, and Control Conference*. Kissimmee.
- Zames, G. (1981). Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses. *IEEE Transactions on Automatic Control* (pp. 301-320). IEEE.

Appendix

Monte Carlo Analysis Results: Wind direction 0 degrees from North.

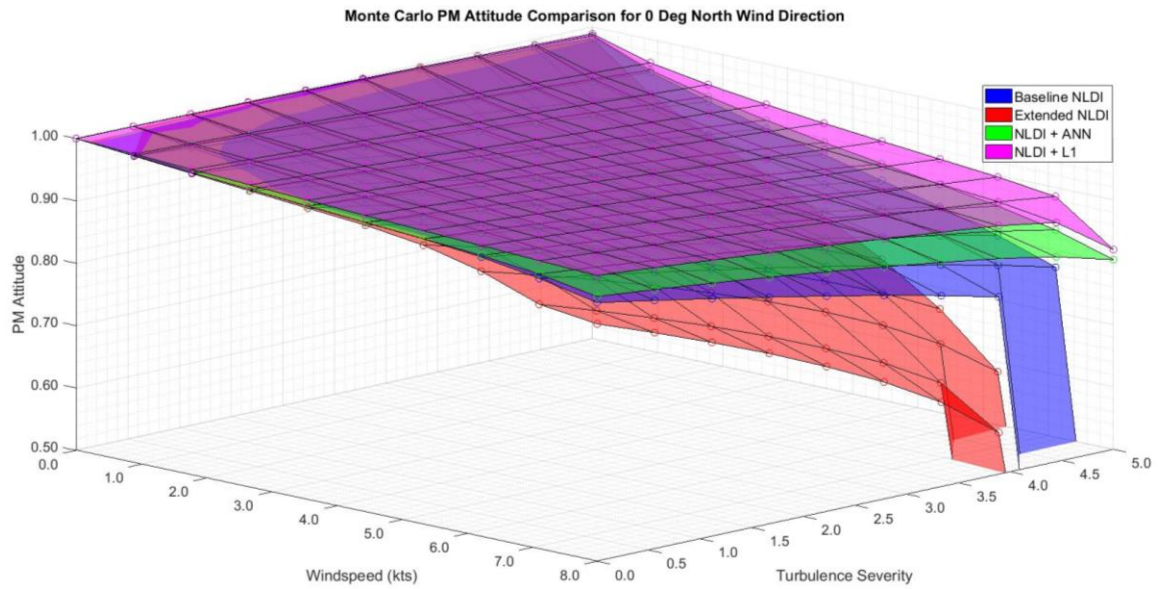


Figure A.1. Controller comparison for attitude angles PM for wind 0 deg from North.

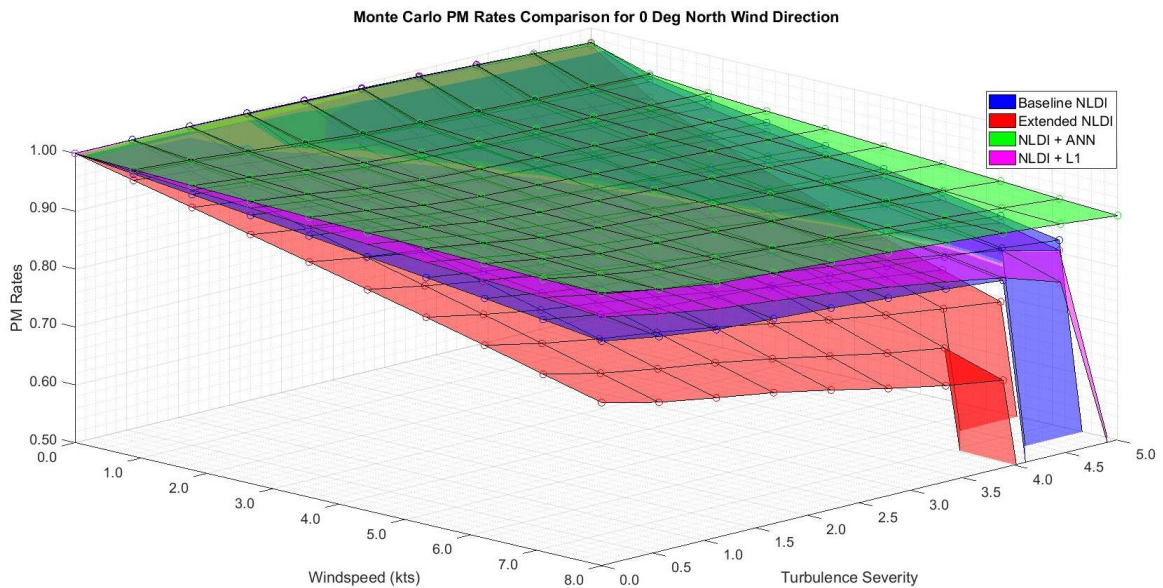


Figure A.2. Controller comparison for angular rates PM for wind 0 deg from North.

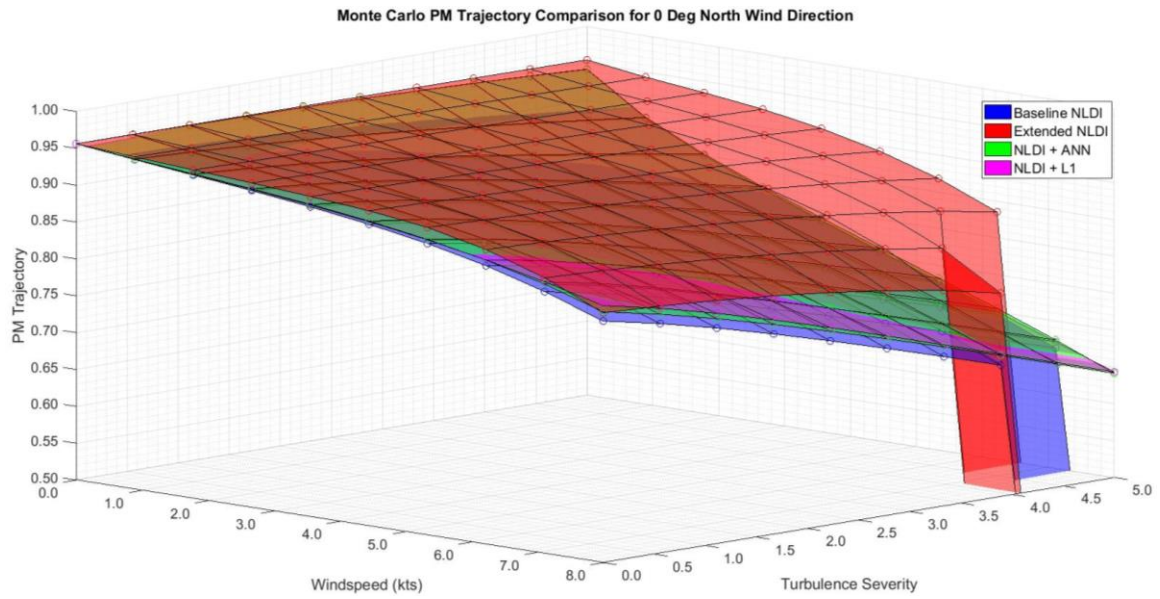


Figure A.3. Controller comparison for trajectory PM for wind 0 deg from North.

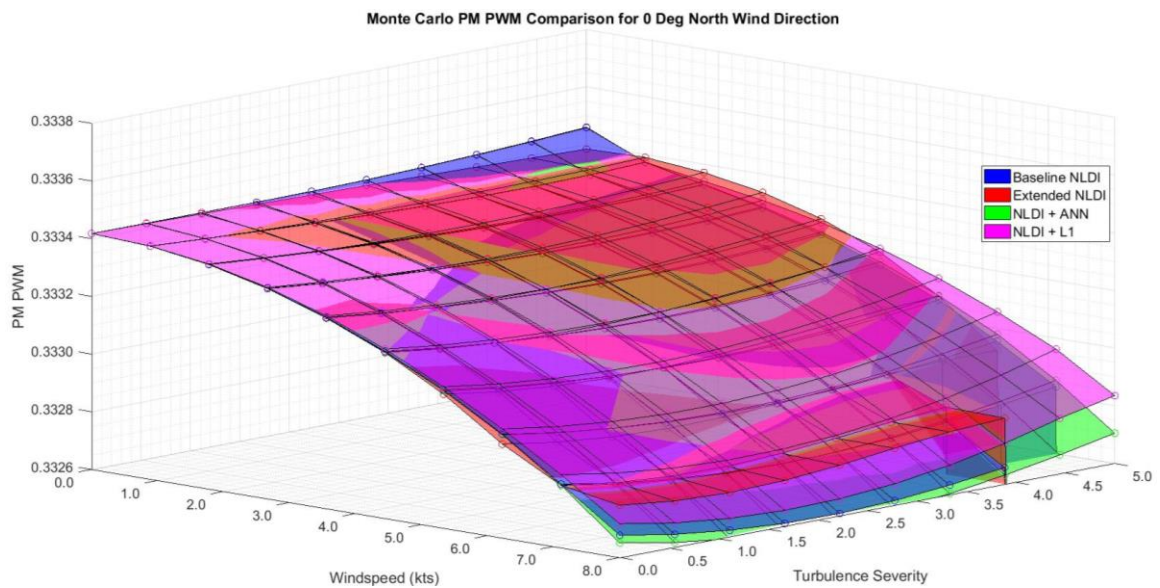


Figure A.4. Controller comparison for PWM PM for wind 0 deg from North.

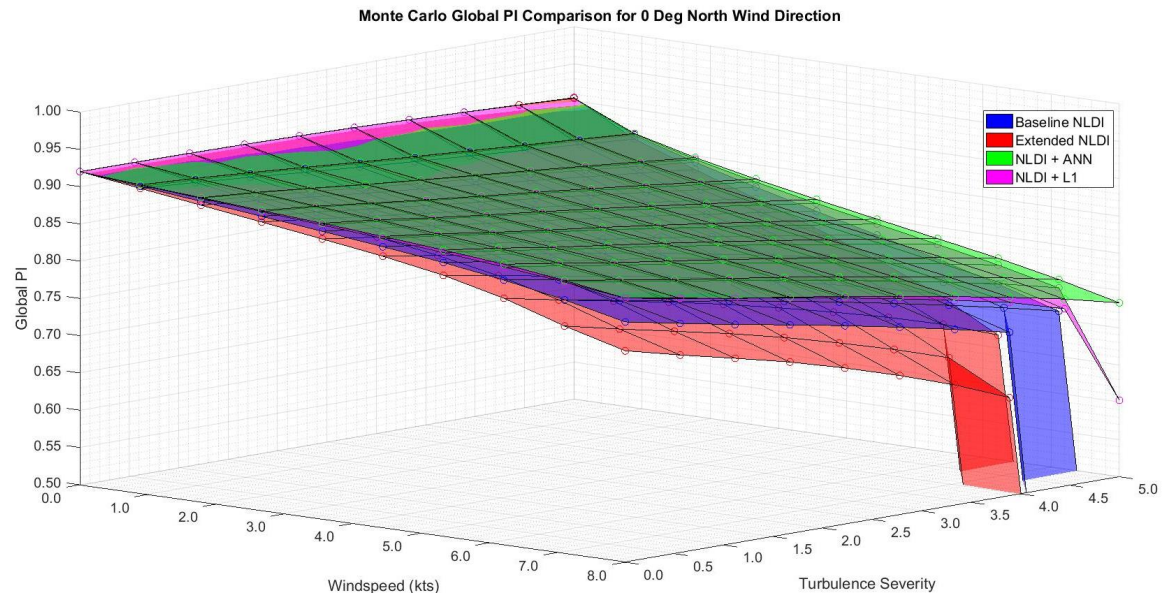


Figure A.5. Controller comparison for global PI for wind 0 deg from North.

Monte Carlo Analysis Results: Wind direction 72 degrees from North.

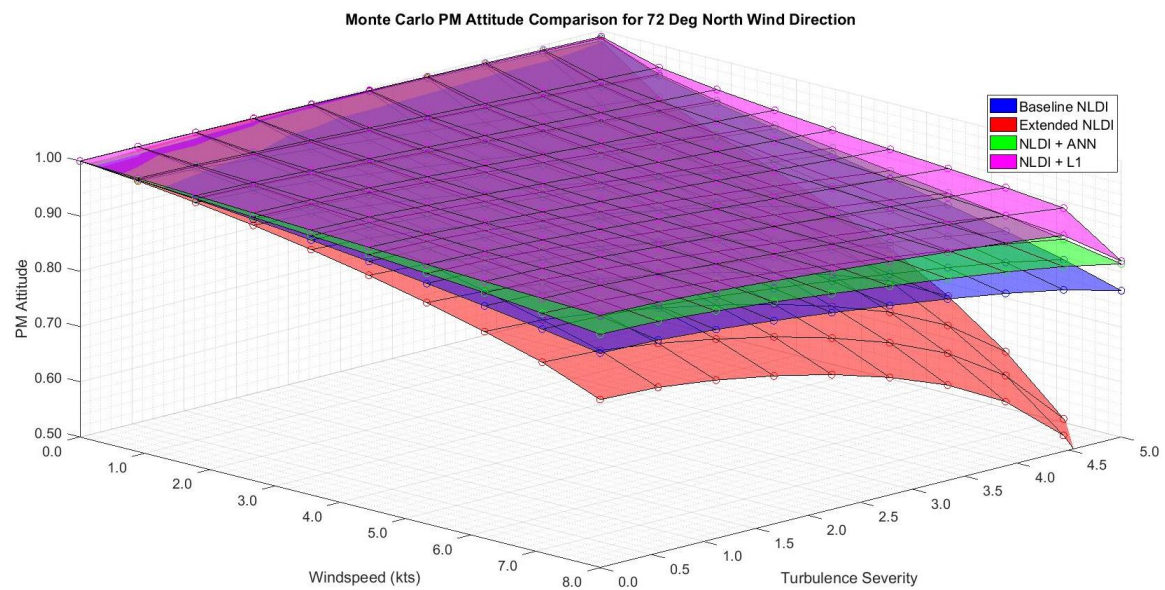


Figure A.6. Controller comparison for attitude angles PM for wind 72 deg from North.

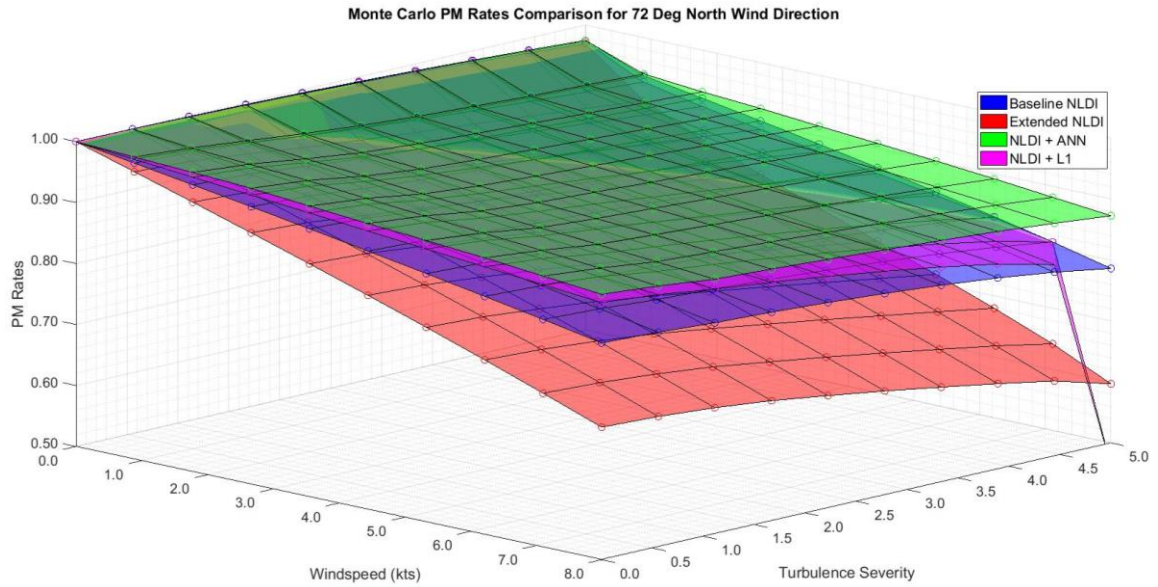


Figure A.7. Controller comparison for angular rates PM for wind 72 deg from North.

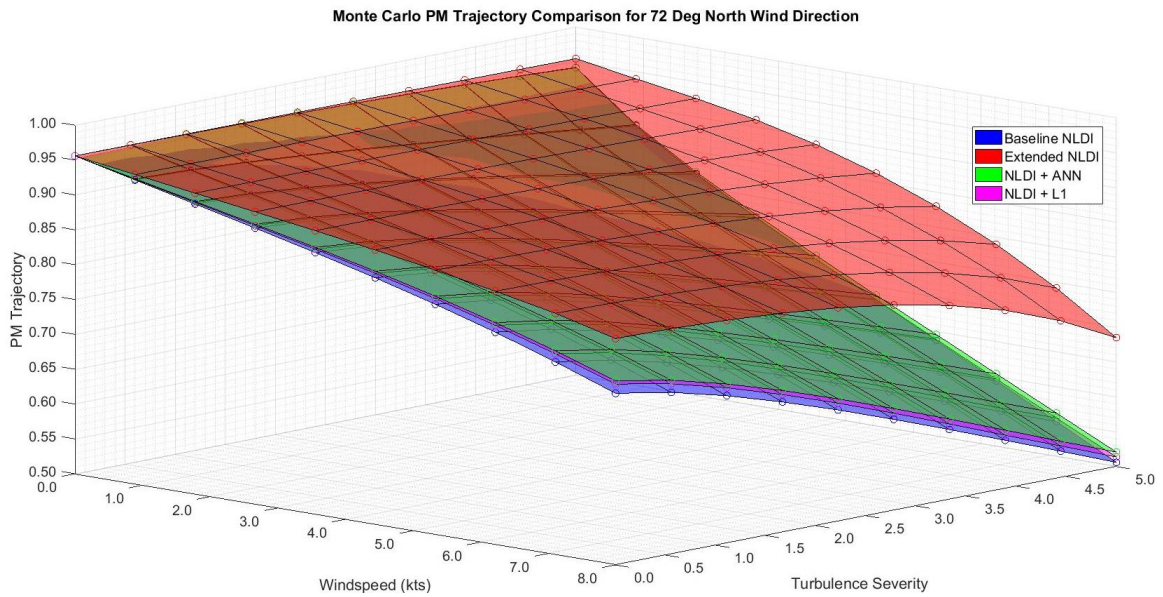


Figure A.8. Controller comparison for trajectory PM for wind 72 deg from North.

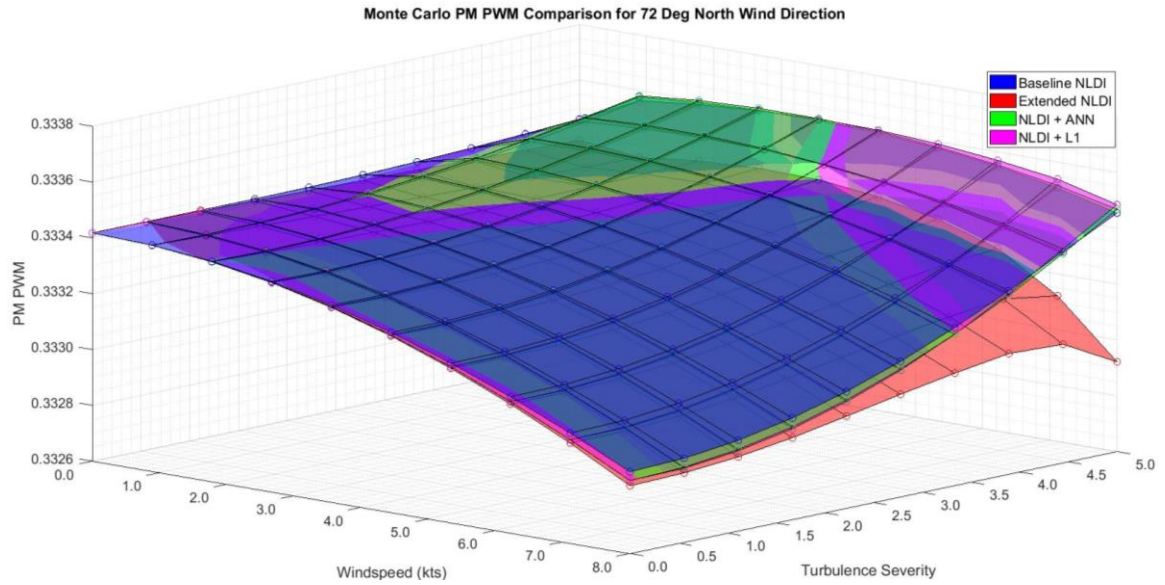


Figure A.9. Controller comparison for PWM PM for wind 72 deg from North.

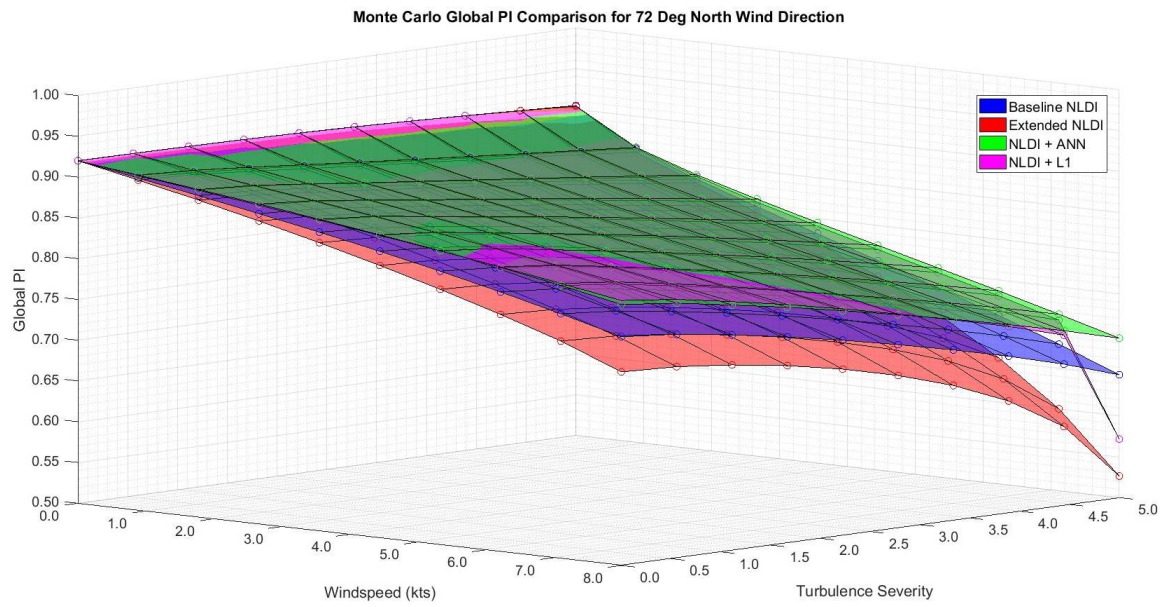


Figure A.10. Controller comparison for global PI for wind 72 deg from North.

Monte Carlo Analysis Results: Wind direction 144 degrees from North.

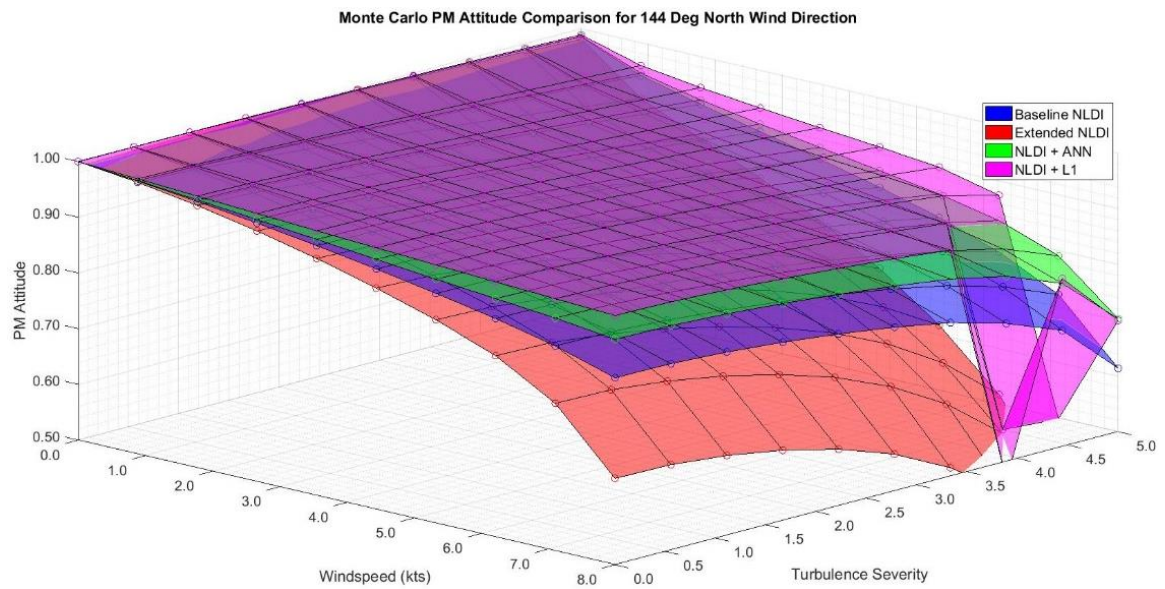


Figure A.11. Controller comparison for attitude angles PM for wind 144 deg from North.

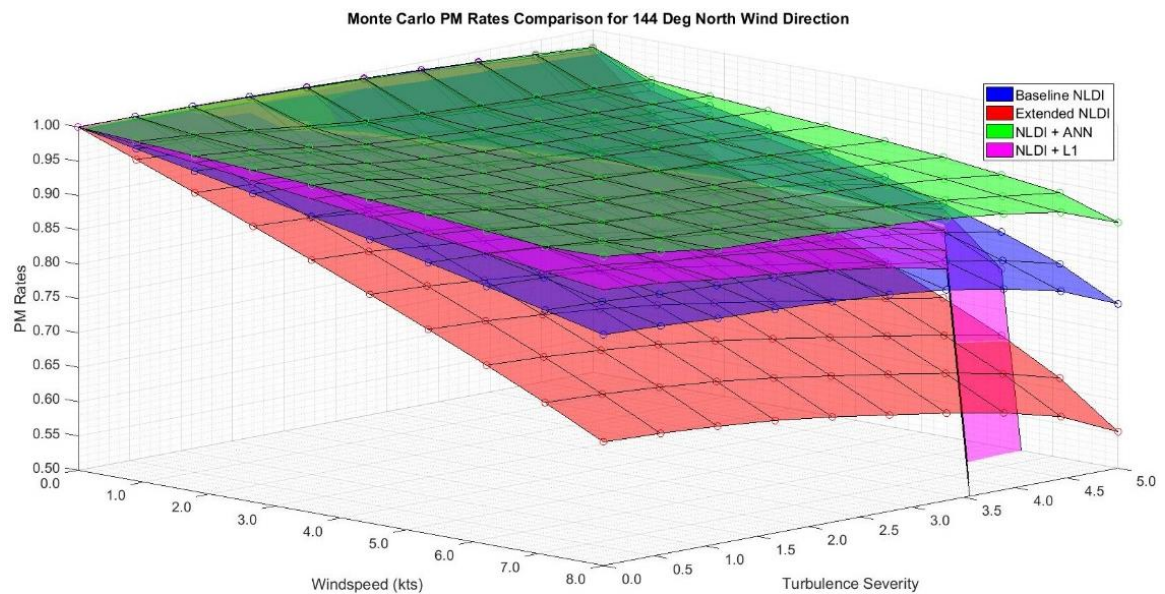


Figure A.12. Controller comparison for angular rates PM for wind 144 deg from North.

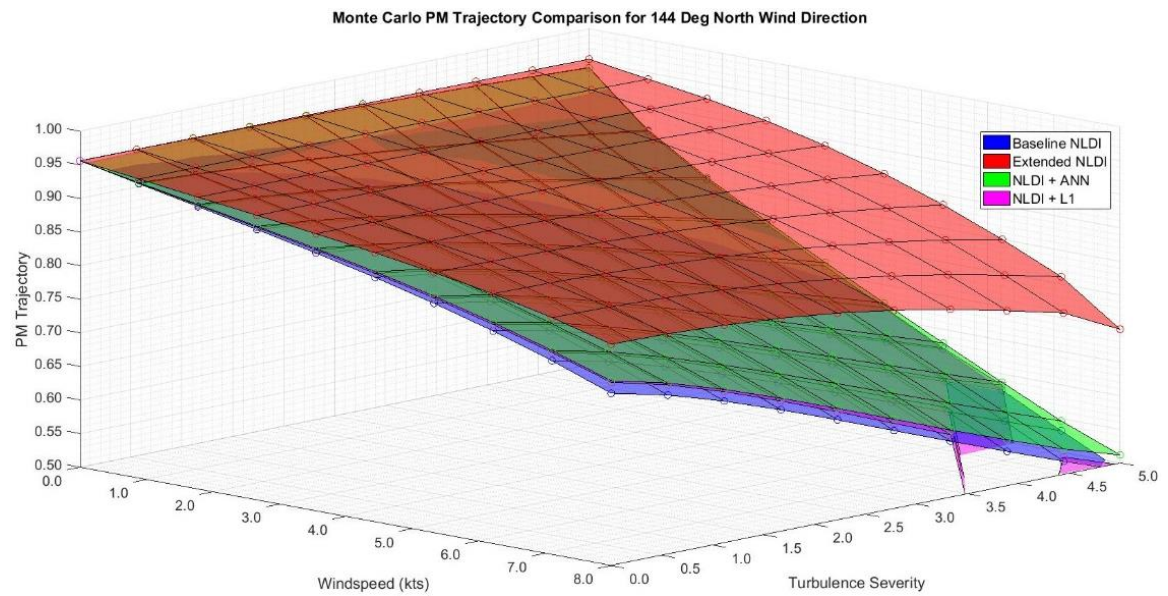


Figure A.13. Controller comparison for trajectory PM for wind 144 deg from North.

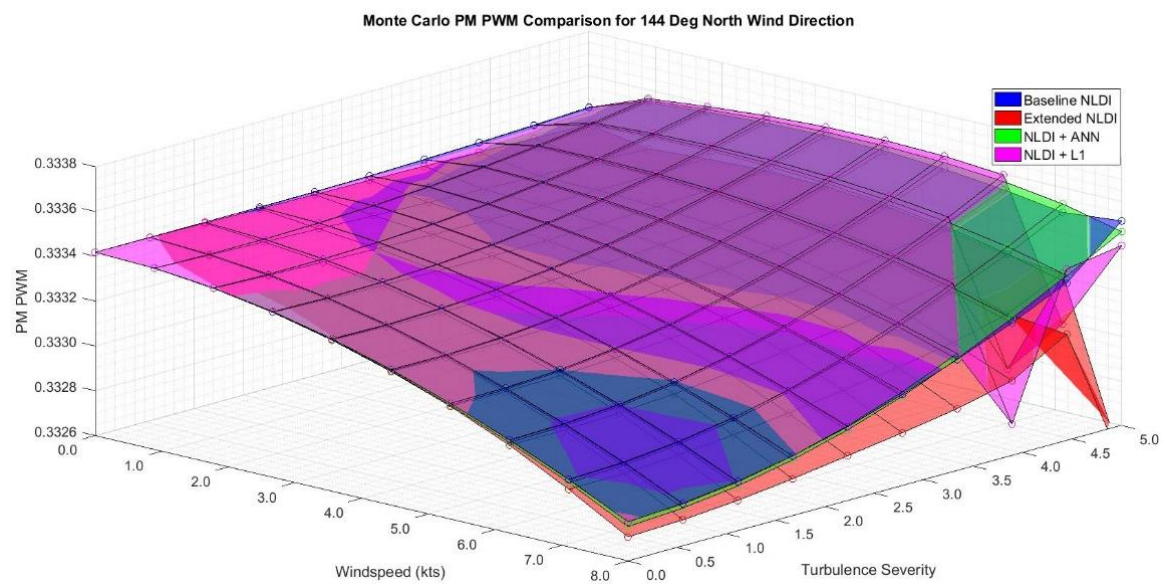


Figure A.14. Controller comparison for PWM PM for wind 144 deg from North.

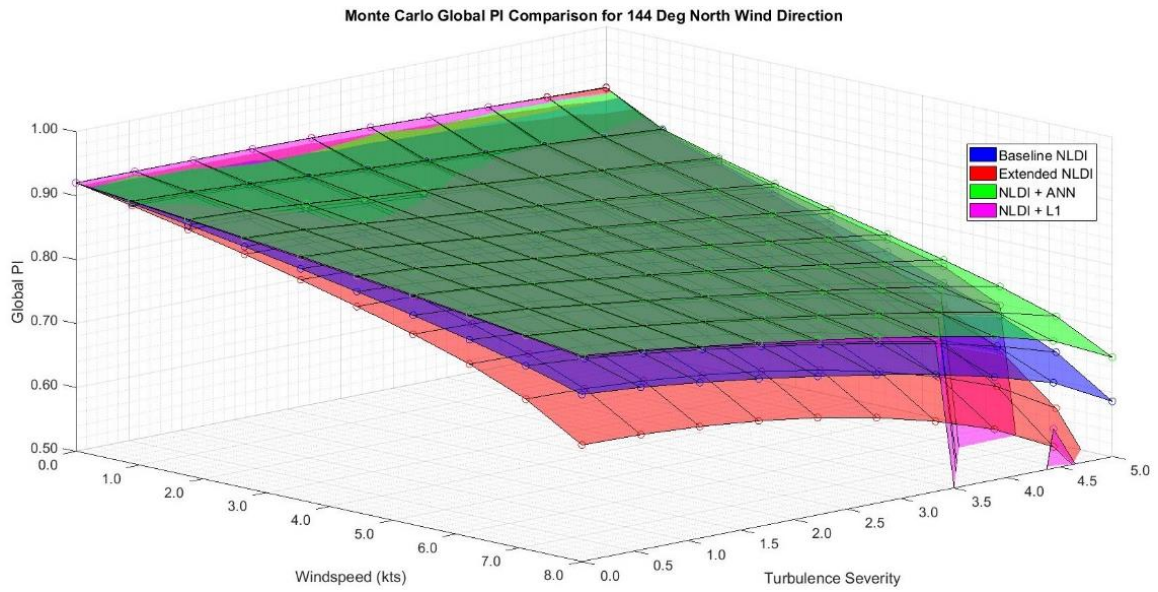


Figure A.15. Controller comparison for global PI for wind 144 deg from North.

Monte Carlo Analysis Results: Wind direction 216 degrees from North.

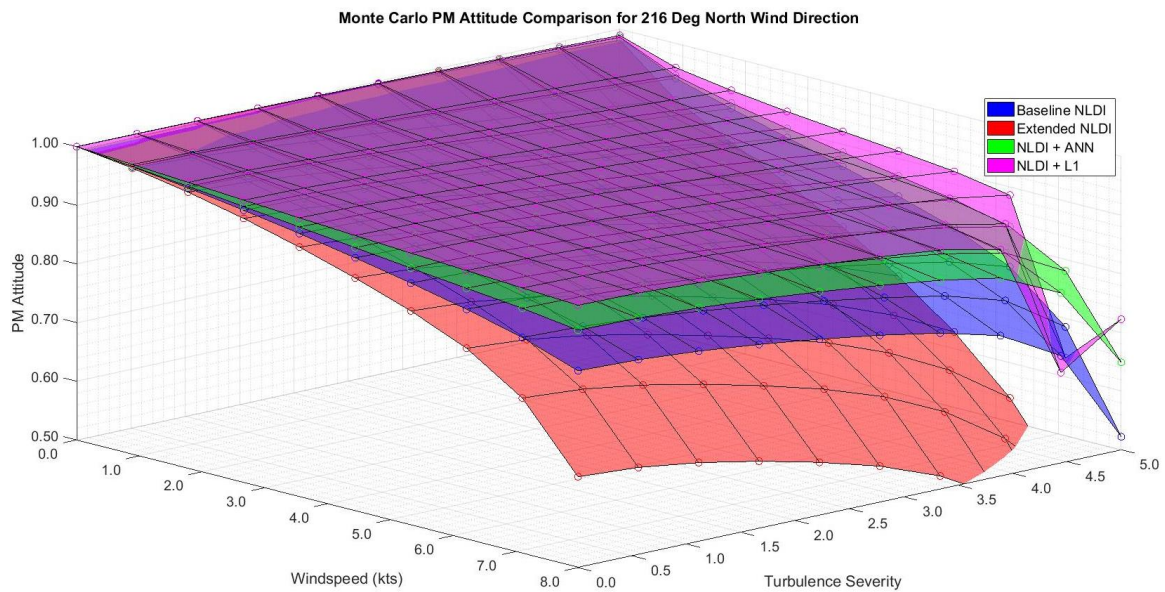


Figure A.16. Controller comparison for attitude angles PM for wind 216 deg from North.

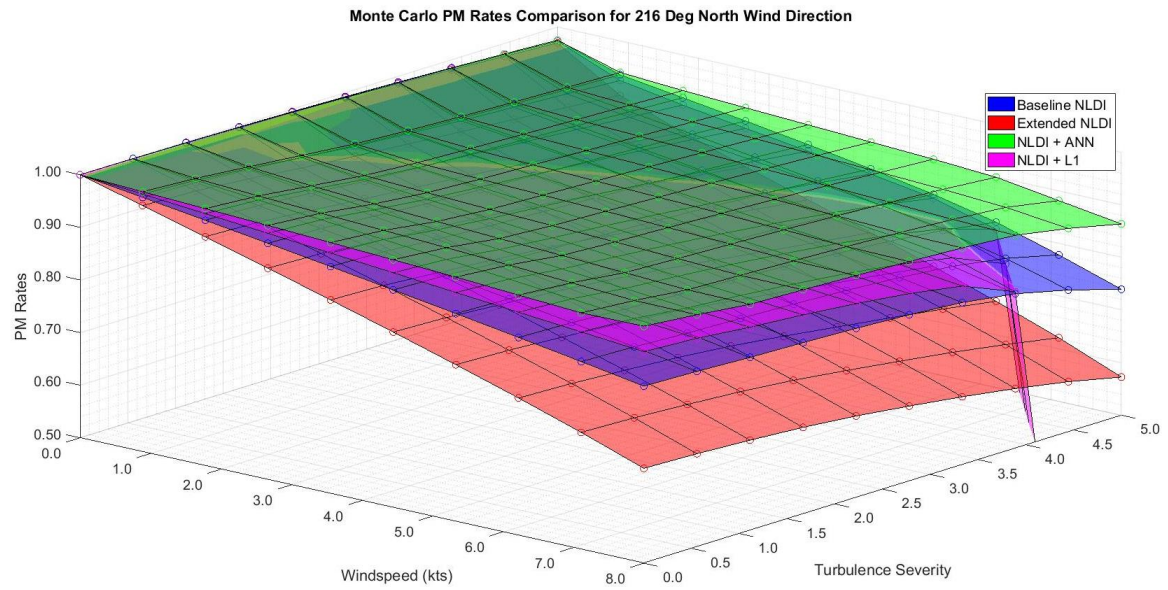


Figure A.17. Controller comparison for angular rates PM for wind 216 deg from North.

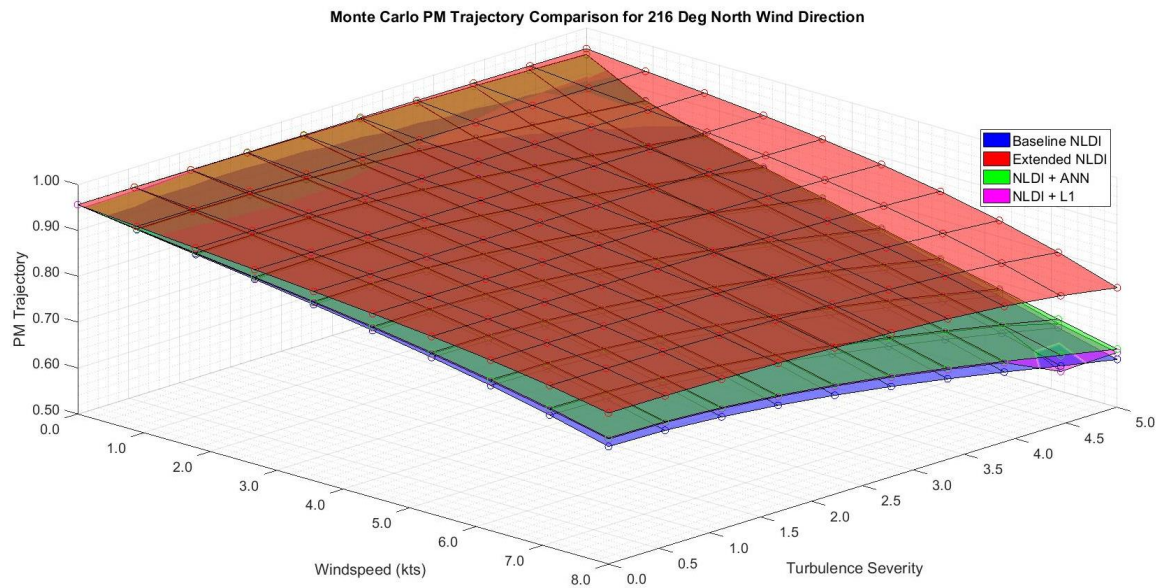


Figure A.18. Controller comparison for trajectory PM for wind 216 deg from North.

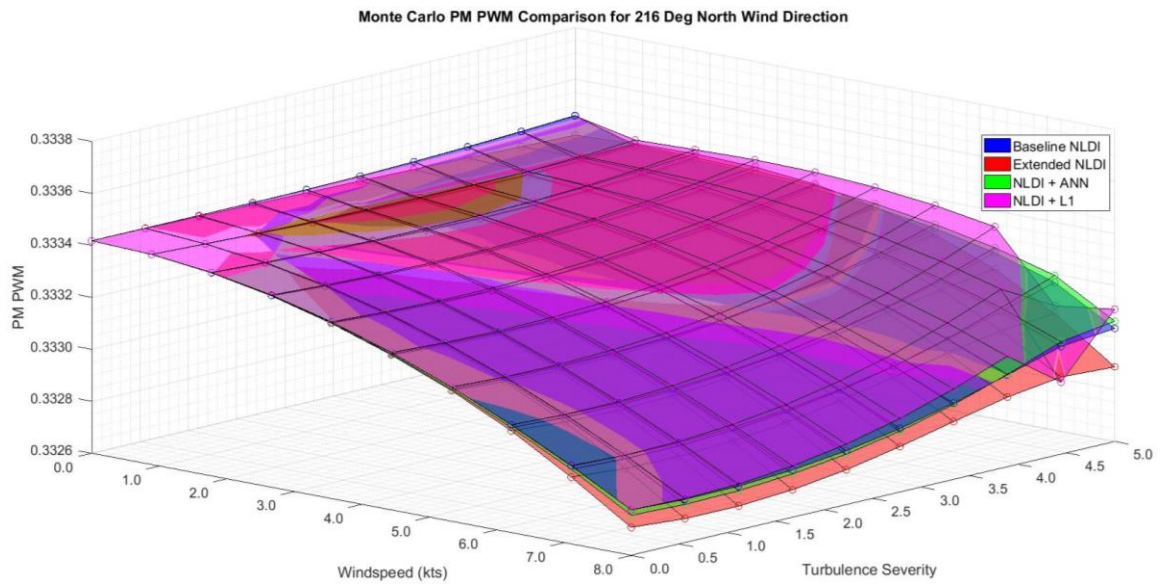


Figure A.19. Controller comparison for PWM PM for wind 216 deg from North.

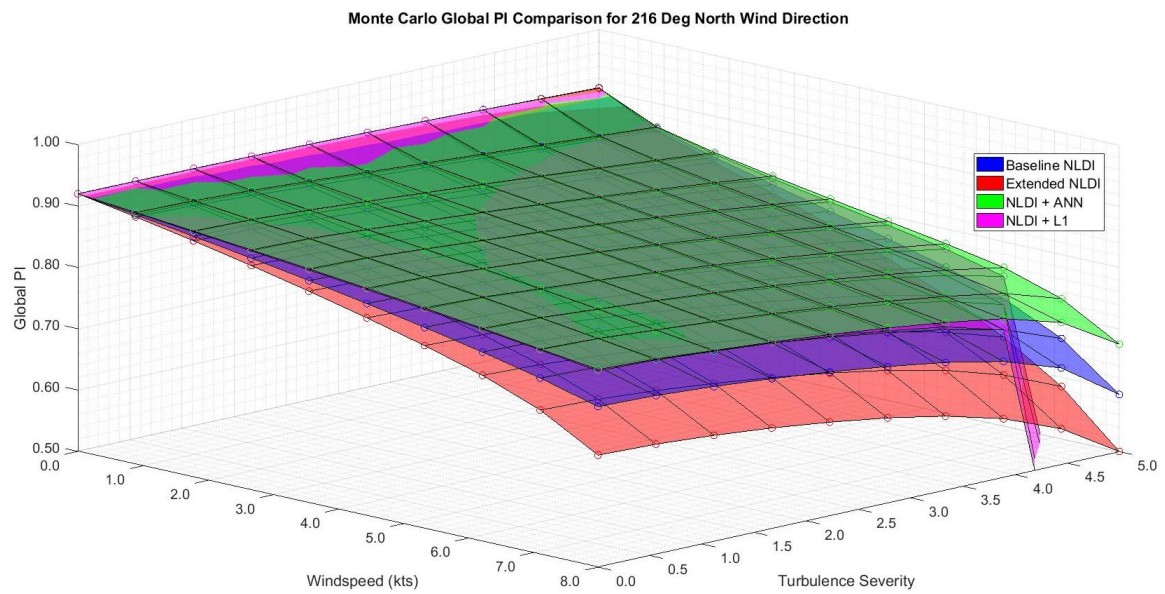


Figure A.20. Controller comparison for global PI for wind 216 deg from North.